AUVSI - 13 YEARS UNDER THE SEA

SERVO MAGAZINE

FOR THE ROBOT INNOVATOR

www.servomagazine.com

September 2010

HOVER BOTS

Wheels? We Don't Need No Stinking Wheels!!

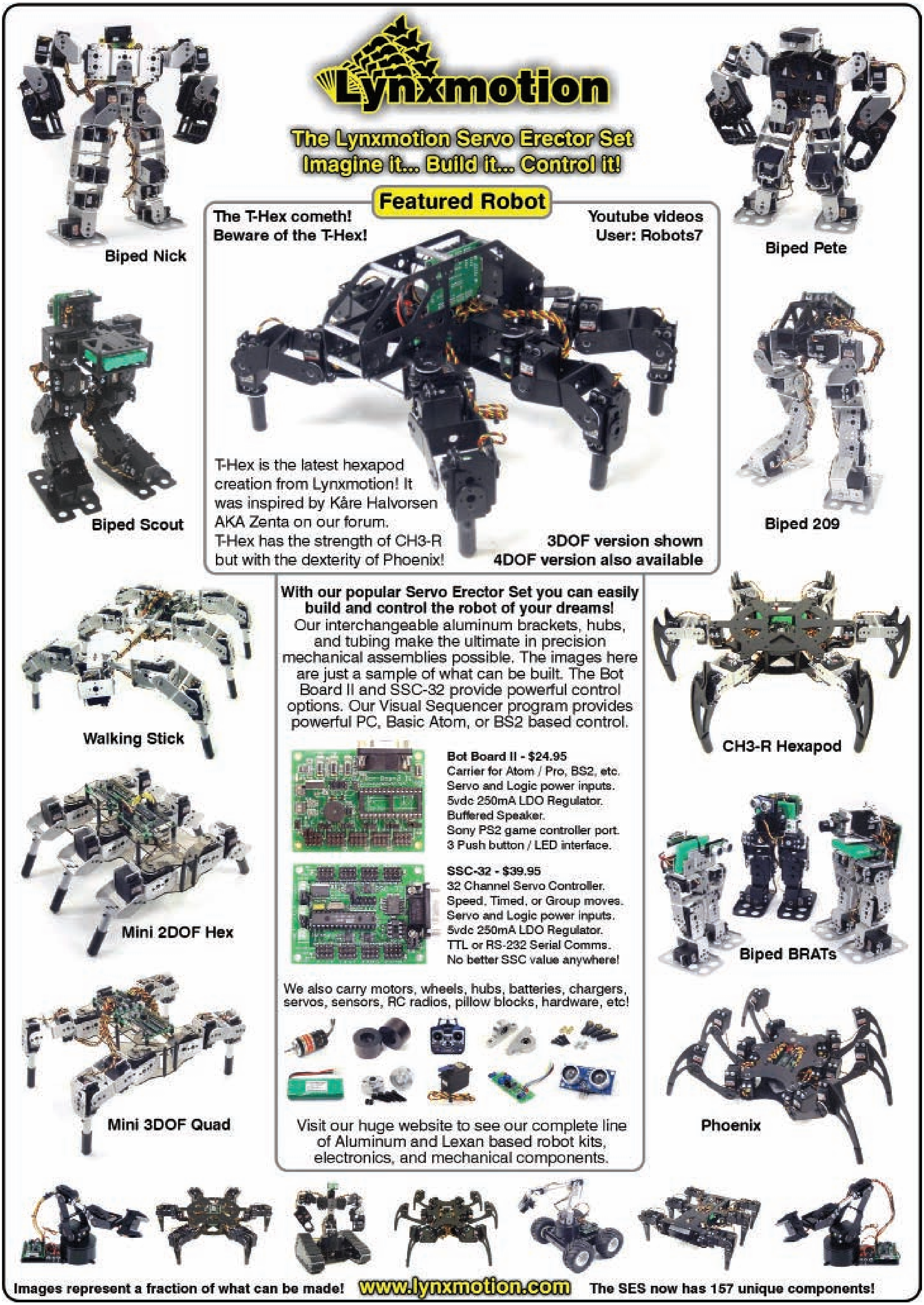
✦ GEERHEAD

TITAN - Guitar Hero Playing Robot From Lee's Summit West HS TITANIUM Robotics Team

U.S. $5.50 CANADA $7.00

09

0 71486 02422 4


Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
Featured Robot
The T-Hex cometh!
Beware of the T-Hex!
Youtube videos
User: Robots7
T-Hex is the latest hexapod creation from Lynxmotion! It was inspired by Kåre Halvorsen AKA Zenta on our forum.
T-Hex has the strength of CH3-R but with the dexterity of Phoenix!
3DOF version shown
4DOF version also available
Biped Nick
Biped Pete
Biped Scout
Biped 209
With our popular Servo Erector Set you can easily build and control the robot of your dreams!
Our interchangeable aluminum brackets, hubs, and tubing make the ultimate in precision mechanical assemblies possible. The images here are just a sample of what can be built. The Bot Board II and SSC-32 provide powerful control options. Our Visual Sequencer program provides powerful PC, Basic Atom, or BS2 based control.
Walking Stick
CH3-R Hexapod
Bot Board II - $24.95
Carrier for Atom / Pro, BS2, etc.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
Buffered Speaker.
Sony PS2 game controller port.
3 Push button / LED interface.
SSC-32 - $39.95
32 Channel Servo Controller.
Speed, Timed, or Group moves.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
TTL or RS-232 Serial Comms.
No better SSC value anywhere!
Mini 2DOF Hex
Biped BRATs
We also carry motors, wheels, hubs, batteries, chargers, servos, sensors, RC radios, pillow blocks, hardware, etc!
Mini 3DOF Quad
Phoenix
Visit our huge website to see our complete line of Aluminum and Lexan based robot kits, electronics, and mechanical components.
Images represent a fraction of what can be made!
www.lynxmotion.com
The SES now has 157 unique components!

SERVO MAGAZINE

09.2010

VOL. 8 NO. 9

PAGE 11

Departments

06 Mind/Iron

20 Events Calendar

21 Showcase

22 New Products

26 Bots in Brief

64 Menagerie

74 *SERVO* Webstore

82 Robo-Links

82 Advertiser's Index

Columns

08 Robytes
by Jeff Eckert
Stimulating Robot Tidbits

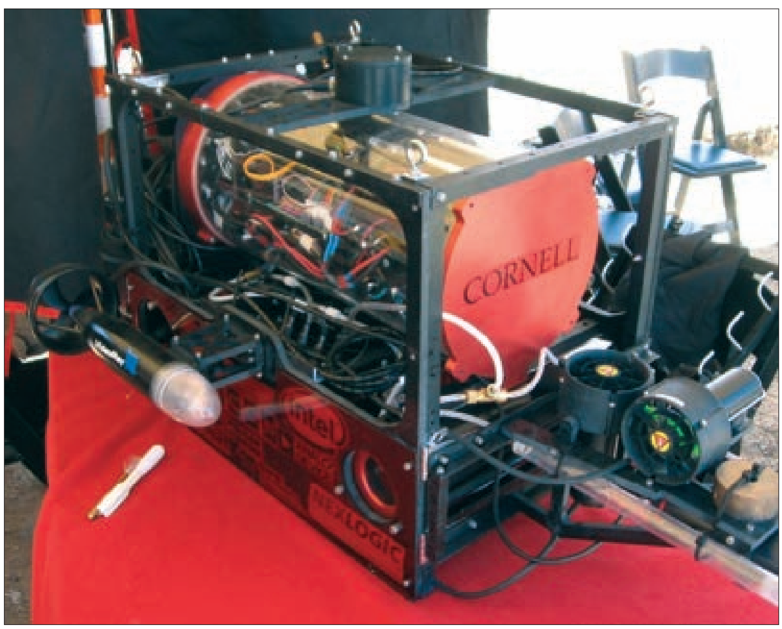
11 GeerHead
by David Geer
Titan, Baddest Guitar Hero Playing Humanoid on the Planet

15 Ask Mr. Roboto
by Dennis Clark
Your Problems Solved Here

67 Twin Tweaks
by Bryce and Evan Woolley
13 Years Under the Sea

77 Then and Now
by Tom Carroll
Robot Manipulators

PAGE 67



SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

In This Issue ...

PAGE 35

The Combat Zone...

Features

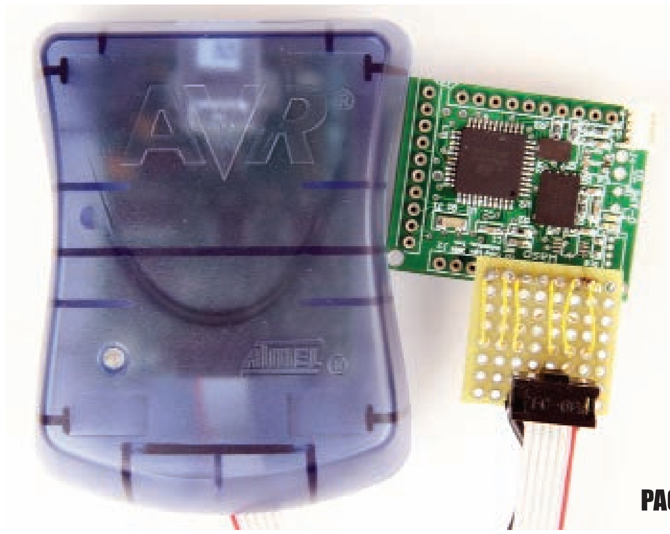
32 PARTS IS PARTS:
Let's Roll — Wheels for Combat Robots

35 MANUFACTURING:
Creating Composite Combat Bots

38 Combat Zone's Greatest Hits

Events

38 Results/Upcoming Events



PAGE 52

40 **The NXT Big Thing #2**
by Greg Intermaggio
This time we get touchy-feely by first learning how the computer software included in the kit works. Then, we'll use our first sensor to navigate our robot around a room.

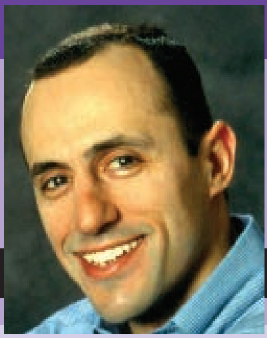
47 **Tool Time With the Versapak Smart Charger**
by Richard Spelling
Whether you have Versapak tools or not, this circuit can be used to charge any NiCd or NiMH battery pack and can restore what you may have thought was lost (battery wise, that is).

52 **The Wasp Embedded Processor Takes the Sting Out of C**
by Fred Eady
If you're already a C expert, you'll be interested in the hardware discussed here. If you're a hardware guru and want to know more about how C and an AVR microcontroller can work together for your robotic projects, there's something for you too.

59 **"Roll" With the HoverBot**
by L. Paul Verhage
Wheels? We don't need no stinking wheels with this unique bot that's lighter than air.

Mind / Iron

by Bryan Bergeron, Editor



Motor Doping

If you're into competitive cycling, you know that the buzz for the past few months has been around so-called motor doping – the use of hidden motors to propel racers effortlessly across the finish line. Whether or not the International Cycling Union uncovers proof of wrongdoing by international racers, the story brings to the forefront the state of the art in battery-powered motors.

According to some claims, the bicycles involved in motor doping had a motor connected to the bottom bracket, and the motor body and batteries fit within the tubes of the frame. Ignore, for the moment, how you'd charge the batteries or mount the motor within a bike tube so that it would be functional, as well as serviceable. The implications for compact or stealth robotics are intriguing.

Given tiny, powerful motors and matching batteries, think of the applications where visible assistance is unwanted or at least not a plus – think wheelchairs and other assistive devices. Then, there's the assistive exoskeletons that could help a soldier carry large loads in the field or help an elderly person manage a set of stairs while carrying a bag of groceries.

An issue is, of course, cost. And that's the beauty of the current situation in cycling, assuming the reports are correct. If the technology is proven on the hills and flats of Europe, it's good enough for cyclists from Amsterdam to San Francisco. Assuming millions of bicycles will be sold with built-in electric assist motors, the price for the technology will be affordable by definition. What a robotics experimenter's dream to be able to walk into a bike shop and pick up a controller, motor, and batteries for a project!

I'm certain the technology exists. I've seen super-compact motors and matching batteries in prototype assistive devices from Japanese robotics shops. However, the demonstrations never last for more than perhaps 30 seconds. If you want to see the state of the art only two years ago, check out "Honda Walking Assist Device" on YouTube. Not something that I'd want to be seen wearing while shopping or just walking around the block. Now, imagine the same device with less obvious motors and batteries. Given the choice between a wheelchair and a not-too-obvious robotic walking assistant, I'd probably go with the assistant.

Of course, raw motor power is necessary but insufficient to make prototypes such as the Honda possible. It's the integration of smart sensor technology with motor control that essentially amplifies the wearer's motor functions. It's one thing to simply press a button to kick a bicycle into motor drive and another to have a device sense when you're applying more pressure to a surface or that the electrical activity in a muscle is changing.

So, again, I'm hoping that the motor doping stories are true, because I want to get my hands on a set of motors and batteries. Then, all we need is a movement akin to either the open source Arduino in microprocessors and shields, or the closed Apple model with thousands of Apps, but focused on the human-robot interface. Whether such a movement originates from a publisher, academic institution, company, or individual, it'll be a bright day for robotics. And, of course, if you happen to be that individual, then get to it. Given the wave of baby boomers that will be hitting their 70s in the next decade, there will be more than enough demand for your technology to keep you profitably occupied. SV

FOR THE ROBOT INNOVATOR
SERVO MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Jenn Eckert, Tom Carroll, David Geer, Dennis Clark, R. Steven Rainwater, Fred Eady, Kevin Berry, Greg Intermaggio, Sean Canfield, Paul Verhage, Richard Spelling, James Baker, Bradley Hanstad, Bryce Woolley, Evan Woolley

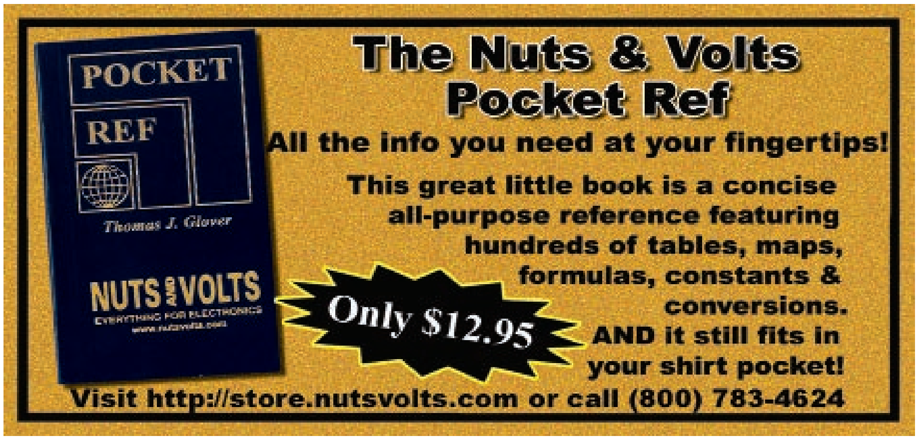
CIRCULATION DIRECTOR
Tracy Kerley
subscribe@servomagazine.com

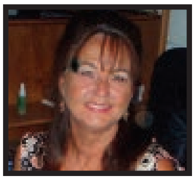
MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Shannon Christensen



Robytes

by Jeff and Jenn Eckert

Taking The Fun Out Of Drowning


Lifeguards vs. Emily: A tough choice?

What do these two photos have in common? Well, mostly, they depict red, bulging subjects from whom you will never receive mouth-to-mouth. However, only one of them shows a real lifeguard. The homely one — dubbed Emily (Emergency Integrated Lifesaving Lanyard) by its creator, Hydronalix (**www.hydronalix.com**) — is a remote-controlled watercraft that can skip out to a distressed swimmer at speeds up to 40 mph and drag him/her back to shore. Originally designed to enable NOAA staff to perform health checks on marine animals, it also carries a sonar device that allows it to scan for subsurface movements such as, oh, flailing around while drowning. In addition, Emily has a speaker system that allows the controlling human to bark instructions to the victim. Her measurements are 54 x 16 x 8 inches; she weighs in at 25 lb (11.3 kg); and, according to Hydronalix, Emily can travel up to 80 miles on a charge. At patrol speed (5 mph), she can operate continuously for a bit more than an eight-hour shift. Next year, a fully autonomous version is planned, to be priced at a mere $3,500. Suddenly the job security picture for lifeguards doesn't look very positive.

Wi-Fi Enabled Trashbot

Also, thankfully, funded by European Union cash is a $3.9 million research program aimed at developing autonomous bots — called "DustCarts" — which are intended to go door to door and collect trash. According to Dr. Paolo Dario, coordinator of the Tuscany-based project and professor of biomedical robotics at the Scuola Superiore Sant'Anna (**www.sssup.it**) in Pisa, "Yes, it is a bin on wheels — there's the drawer in which you place your bag of rubbish or recycling — but there's a lot more to the robot than that." Indeed, as it has been fitted with cameras so it can see where it's going and avoid obstacles like broken-down Fiats and opera tenors, and it uses Wi-Fi triangulation to find its way around inside your home. However, these things are basically just wastebaskets mounted on a Segway, so it seems like you might need quite a few of them for anything larger than a tiny village.

In a two month test in Peccioli, Italy, two DustCarts managed to take care of about 100 households. This means that it would take about 25,500 of them to service Los Angeles. If you think traffic is bad now, just wait until these things get out on the Santa Monica freeway.



DustCart, the European Union's robotic trashman.

Curiosity On Wheels

As widely reported, Mars rovers Spirit and Opportunity are still on duty. The former is currently stuck in the sand and sleeping on the job, but the latter is still plodding along on its journey to the Endeavor Crater. It's all very interesting, but given that they are six years into a planned 90-day mission, it may be a good time to think about the next generation: Curiosity.

Mars rover Curiosity is being assembled for testing prior to its late 2011 launch. Photo courtesy of Nasa/JPL-Caltech.

The new rover is currently under assembly and will undergo prelaunch testing over the next few months. Curiosity is a six-wheeler that sports the same rocker-bogie suspension as its predecessors which eliminates axles and springs and allows a vehicle to climb over obstacles that are up to twice the size of the wheel's diameter while keeping all six of them on the ground. Each wheel has its own drive motor, and the corner ones also sport independent steering motors. Unlike the previous models, this guy will use his own suspension as landing gear when lowered to the surface on a tether.

Curiosity's mission will begin with a Florida launch late next year and, although it isn't specifically looking for signs of life, will involve creeping around an area of Mars that may once have been habitable. It will examine rocks, soil, and the atmosphere with an array of tools, including a laser for vaporizing rocks from a distance and equipment designed to test for organic compounds. We'll just have to wait and see if it, too, stays on the job 24+ times as long as intended. You can follow its progress at **www.nasa.gov/mission_pages/mer**.

Flying Bugs Bounce Back

There's a lot to be said for paying grown adults to play around with toys, especially when their paychecks are covered by someone other than US taxpayers. Researchers at Switzerland's Ecole Polytechnique Federale de Lausanne Laboratory of Intelligent Systems (**lis.epfl.ch**) are engaged in a study of the physical interaction between indoor flying robots and their environment, specifically in terms of surviving collisions with other objects while in flight. Whereas most air vehicles must incorporate sophisticated collision-avoidance systems, the Airburr is designed to survive violent encounters. This appears to be accomplished mainly by encircling the flyer in a springy protective cage, but the concept also includes smart sensing to detect the collision force and location on the bot and to react so as to avoid mission-compromising damage.

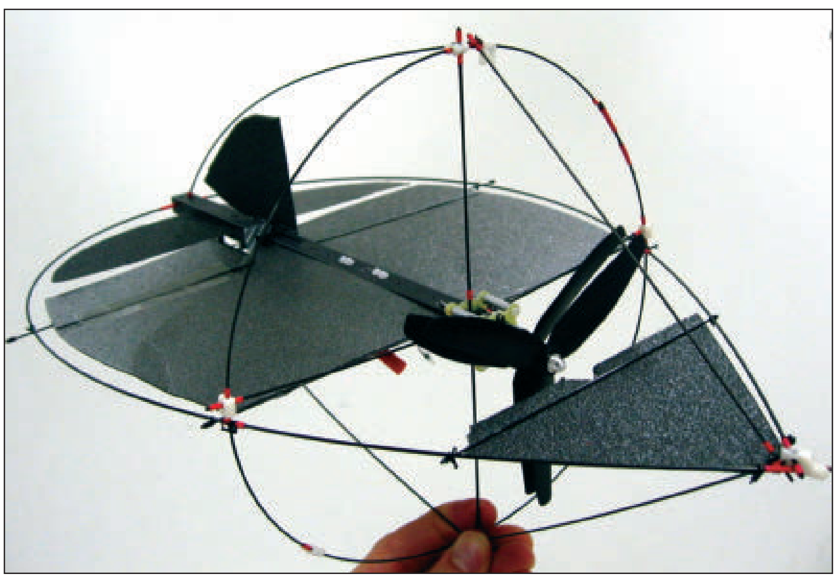
The prototype shown employs carbon fiber rods, mylar foil construction, and lightweight electronics to keep it light. Careful attention to the center of gravity allows it to fly in either forward or hover mode. According to the literature, "Instead of fearing contact, the AirBurr will take advantage of its environment. It will be able to attach to surfaces to have a bird's-eye view of a room without using energy for lift ... It is hoped that these innovative features will bring us closer to mimicking the impressive flying characteristics seen in insects and to understand the mechanics and control required to create a truly useful and robust indoor flying robot."

Now, where's my fly swatter?

Bot Baby To Stimulate Reproduction?

He's round, soft, makes baby noises, kicks when you

The Airburr prototype collision-proof indoor flying robot.



Yotaro, a possible inspiration to young Japanese couples.

tickle him, and has a permanently runny nose. No, it's not John Goodman, but nice try. It's Yotaro the robotic baby. Originally designed as an experiment in transcultural aesthetics, Yotaro has recently been touted as a possible solution to Japan's dwindling population which has reached such crisis proportions that the government has enacted a $150 per month subsidy for each child a family adds. Supposedly, Yotaro is so dang adorable that all you have to do is touch him, and you'll want to rush home and create the real thing.

"I think it's true that young working couples have no chance to have personal contact with babies in their lives. The people who came to the robot exhibitions enjoyed touching Yotaro, like a real baby," said Hiroki Kunimura, Yotaro project leader at the University of Tsukuba (**www.tsukuba.ac.jp**). "A robot can't be human, but it's great if this robot triggers human emotions so humans want to have their own baby." Haven't these people heard of dirty movies? **SV**

GEERHEAD

by David Geer

Contact the author at geercom@windstream.net

Titan, Baddest Guitar Hero Playing Humanoid On The Planet

The complete Titan Guitar Hero playing robot system with speaker system, television, web camera, DVD media player, the robot with laptop head, hand actuators, and guitar.

A high school robotics team is tasked to solve complex programming and design problems in robotics to achieve a worthwhile goal for their age group in popular culture. Their resolution: a humanoid robot that effectively competes in a significantly high percentile in the Guitar Hero video game, defeating most opponents.

Titan just may be the most "legit" (i.e., coolest) non-commercial robot rocking the Internet today. This music-savvy mechatron has the licks to rock expert level on Guitar Hero, playing songs with the difficulty of Iron Maiden's "The number of the Beast" or Eric Johnson's "Cliffs of Dover."

Decked out with a laptop head, black PVC construction, and brand name chest plate, Titan — named for the mascot of Lee's Summit West High School which produced the robot's creators — was inspired by another Guitar Hero robot that the high school team discovered in an exhibit.

Team Titanium came up with the idea for their robot at the 2009 FIRST Robotics Championships in Atlanta. "As a major FIRST sponsor, National Instruments had a big presence there, including a large booth area with various demos and seminars. All of the FIRST participants there were mesmerized by a very cool project they were showing

The web camera mounted in front of the TV where the notes are displayed.

off: a Guitar Hero playing machine," says Jeff Spatz, head coach of the high school Team Titanium.

The NI system used an industrial machine vision camera, a high speed controller, and a complex pneumatics system to play the plastic guitar as part of the famous video game. The NI robot used the camera to read the notes rolling off the TV screen so the robot could play them — and play them quite well.

Building a robot better than that of NI posed several problems. Cost was a factor, since the high school robotics team operated on a very limited budget. During the first stage of development, the team focused on duplicating the robot's real-time vision task: seeing, transmitting, and translating the notes into instrumentation by the robot's hands.

The team would have to use an inexpensive web cam and a common laptop instead of the costly, high frame rate camera and high speed controller device used by NI. The students were concerned the camera might not be fast enough to pick up all the notes and that the laptop might not be fast enough to process the vision task and control the robot's fingers (guitar actuators).

Fortunately, Team Titanium was able to use NI's LabVIEW industrial control software which helped the team in producing their FIRST competition robots. The software includes video and vision tools and libraries, but would that be enough to create sufficient programming to recognize all the notes and outgoing signals for each note?

A Glimmer of Hope

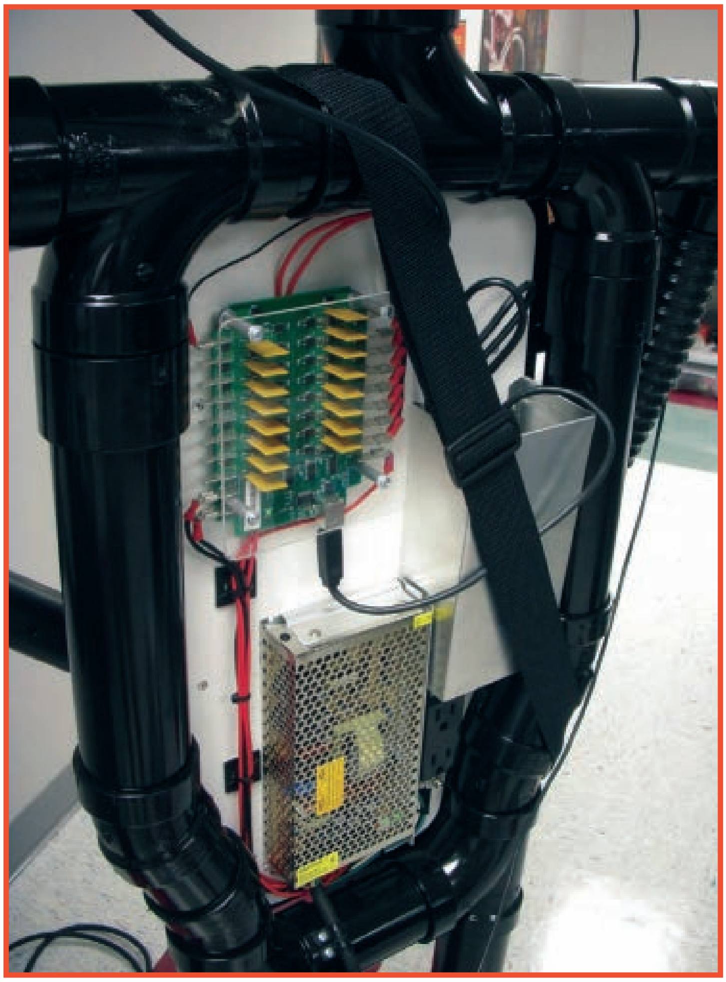
Within a few short hours of tinkering, the team's software writer had a program running. The web cam was watching the TV screen, the game was running, and the program was flashing lights on the LabVIEW interface in tandem with the notes on the screen. "We did not know if the results were good enough to play the game well, but it was clearly good enough that we knew the vision task was within reach," remarked Spatz.

Moving on to the robot's construction, the team quickly decided to create a standing, humanoid robot with arms and legs that would approximate a real person playing the game. It would even be able to compete with real people. This way, the robot would demonstrate the team's capabilities, help recruite future team members, and perhaps even help raise money for the team.

The actuators would need to function similarly to hands and fingers. They would need to attach to the guitar and be connected back into the robot's system. This way, the guitar and strap could hang on the robot just like on a real rock star, making the robot look more lifelike. "We decided to use simple solenoids for low cost and simplicity. We purchased a few 12 volt cylindrical solenoids, and tinkered with various finger designs to actuate the buttons on the guitar neck and toggle the strummer on the guitar. These two tasks were completely different design projects, but we were able to do both with the same solenoids," says Spatz.

The team also wanted the robot to employ the guitar's whammy bar which is similar to the hardware on a real electric guitar used to stretch and release the strings for special sound effects. An

The robot's laptop head runs the LabVIEW-based interface and program for playing Guitar Hero.



The robot's power supply, relay board, and control panel on the back of the robot.

additional actuator was adapted to that task.

"Using the Autodesk Inventor CAD that we used to design our competition robots, we designed hand assemblies that matched what we learned from our prototypes, and started to fabricate hand parts. After much tinkering, we had two hand mechanisms that could attach to and operate the guitar," Spatz explained.

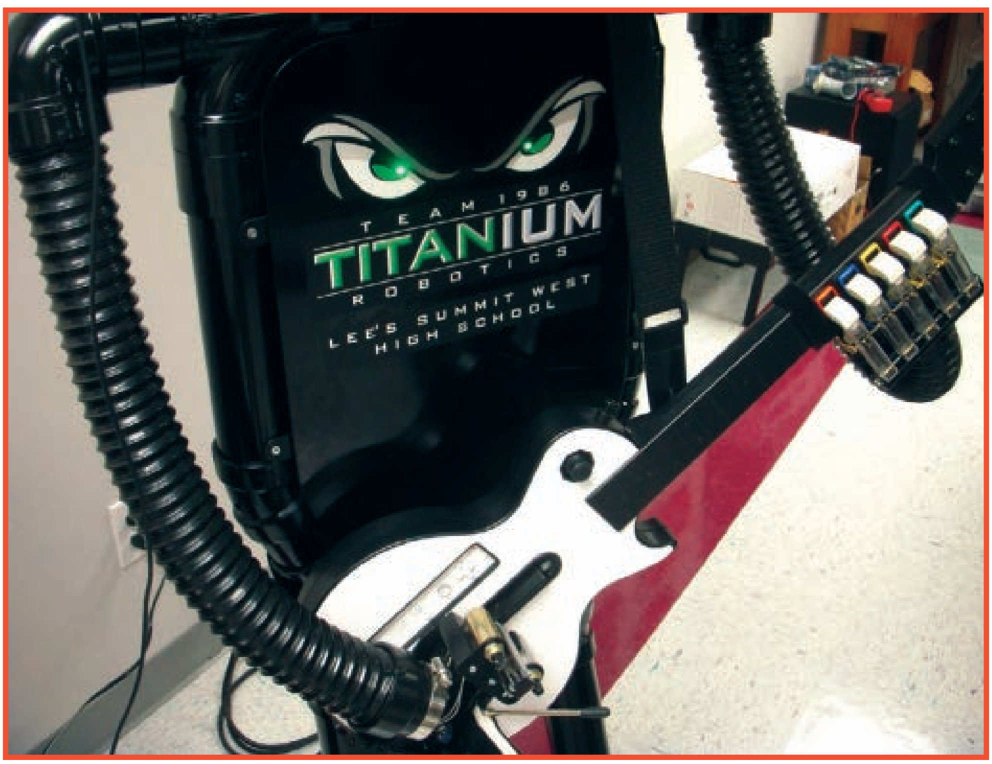
Once the vision and control software were completed on the laptop, and the hand mechanisms were ready to play the guitar, the team simply needed to create an I/O link between them. "We were hoping to find an inexpensive, all-in-one solution to go directly from the USB port on the laptop to the 12V high current signals we needed to drive our solenoids (the hand actuators). After much searching, we found it at a small irrigation controls company on the web called Bibaja," Spatz continued.

Bibaja's I/O solution used a small, cheap controller board designed for controlling irrigation systems. It translated the USB data into 16 channels of 12 volt output, Spatz commented. "We got the board, wired up the system, and were quickly playing Guitar Hero even better than we expected. There was still much experimenting and improving to do, but we knew we were in the ballpark," Spatz conveyed.

Using multiple variables in the programming, the team was able to make many adjustments. They were able to adjust the screen size, the locations for the camera and system to watch, the brightness level and duration of the light required to indicate a note, the ranges that occurred between notes, the time delay allowable between seeing a note and playing it on the guitar, the length of the output signal to the solenoid actuators and the length of other signals, and the length of the note required to activate the whammy bar, listed Spatz. With hours of practice and adjustment, they arrived at the best settings for the robot to play at the most expert level that it could.

Body and Soul

Next, the roboticists constructed Titan's body on which they mounted everything. They made it out of PVC pipe, with a glossy black



Close-up of the robot as its actuators play the guitar.

Close-up of the right hand strumming the guitar.

Close-up of the left hand fingering the fret board.

coating of paint.

They used vacuum hose to create the flexible arms, so they could connect them to the hand assemblies and use them to hide the wiring.

A 12 volt power supply mounted near the relay board and power outlets on a control panel formed the electronics in the robot's back. "It was clear that the controlling laptop computer should be his head, so we added a platform on which it sat. We added colored LEDs to his fingers which flash as each finger is actuated. This made him much more interesting to watch," Spatz detailed.

To round out the self-sufficient, portable robot system, the team added a Wii console, guitars, a flat screen TV, and a speaker box with car speakers and amplifier, as well as wheels on which it could travel.

The robot currently scores in the 90s in the expert level of Guitar Hero for most songs it plays. Kids love to watch the robot at exhibits and demonstrations. "Many young Guitar Hero aces challenge him, but very few can beat him. We have added the ability to set him to play at any difficulty level (easy, medium, hard, or expert) so that he can play along with any level of challenger. He has become a great ambassador for our team, and quite the celebrity in our community," Spatz mused.

The team has put Titan on display at sports events, science fairs, camps, school assemblies, street fairs, and FIRST Robotics regional tournaments. "He even performed before a crowd of 2,000 Cerner Corporation employees at their annual convention at Kemper Arena in Kansas City. He has become popular on the Web, and we have received comments about him from all over the world," concluded Spatz.

Resources

Titan, original introductory video
www.youtube.com/watch?v=VLQhX0nZV6I

Titan, video playing "Cliffs of Dover" expert
www.youtube.com/watch?v=0mWHVvKb1hM

Titan, featured on Kansas City TV channel
www.youtube.com/watch?v=7uTnSHbRjF4

Titan, performing for Cerner Corporation convention
www.youtube.com/watch?v=zUEVzjbzjD4

Titan, story from school TV channel
www.youtube.com/watch?v=5Yx-JYXEqh8

Kansas City TV feature story about the student who programmed Titan
www.youtube.com/watch?v=iu2Ocj82zQg

National Instruments interviews with the student who programmed Titan
www.youtube.com/watch?v=kYhJMB8irhw&playnext_from=TL&videos=tUbZF0gNknk and **www.youtube.com/watch?v=ievSpmNU1fw&playnext_from=TL&videos=Ca72RhQ7zj4**

NI LabVIEW software used with Titan
www.ni.com/labview

Bibaja's I/O technology
www.bibaja.com/products/index.php?page=iopoint_usb

Final Notes

Titan achieved all expectations as an attractive, highly accomplished humanoid, Guitar Hero playing robot. Team Titan also uses the robot as a fundraiser, charging a small fee to those who want to challenge the robot in Guitar Hero. The robot has been a great addition to the robotics team's PR efforts, striking a deep chord with the community. SV

Our resident expert on all things robotic is merely an email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

by
Dennis Clark

I'm pondering the meaning of this year's First LEGO League (FLL) topic "Body Forward" — a bio-medical robotics topic. What could it mean? How will my team do? What will the pizza cost? Just by coincidence, I received a LEGO NXT question! FLL does not allow teams to use anything but the official Mindstorms programming language, but that doesn't stop you robot builders out there from using it! So, with that in mind, let's get right into the big question this month.

Q. I've heard that there is a way to program the Mindstorms NXT in C without using the LEGO Mindstorms graphical programming language. Can you tell me how?

— Erik

A. There is such a thing, but it isn't really C. It just looks like it. It is called *NXC* which stands for *Not eXactly C*. NXC is basically a set of macros laid over the top of the actual programming language for the NXT brick called *NBC* — an acronym for *Next Byte Codes*. The NBC assembler takes source written in NBC and compiles it to the byte codes used by the NXT brick firmware interpreter. Both NBC and NXC are open source projects that can be found here at **http://bricxcc.sourceforge.net/nbc**.

The project keeper of the keys is John Hansen, according to the **sourceforge.net** project pages. John has a "Power Users" NXC programming book in its second edition that is very well thought of, called *LEGO Mindstorms NXT Power Programming: Robotics in C*. Use it to show yourself the ropes of using the NXC language. There is also an on-line NXC programmer's reference here at **http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/index.html**.

I must admit that I didn't know anything about this language or its usage until I researched it to answer this question. Since I have a couple of Mindstorms NXT kits (I coach my kid's school FLL team), I thought that it would be a great thing to learn. I am a big fan of Dave Baum's MacNQC program that brought *Not Quite C* to the Macintosh "way back when." Unfortunately, no one carried the Mac flag forward for NXC beyond the command line compiler support. If you are a Windows user, you are in luck however, since the well-supported GUI program *Bricx Command Center* has recently come out with NXC support in the 3.3 versions. There is more information and downloads available at **http://bricxcc.sourceforge.net**.

Typically, it's a good idea to download the newest release version first and THEN go get the most recent build or "test" version to update that.

If you are a Mac OS X or Linux user, the command line tools actually work great. As most of you are aware by now, I am a Mac OS X guy, so the following directions will detail how I got NXC working on a Mac OS 10.6 (and 10.5 and 10.4) version.

Step One: Get the NBC compiler.

The very latest NBC/NXC compiler is 1.2.1 r3 (June 30, 2010). Go to **http://bricxcc.sourceforge.net/nbc** and click on "Mac OSX (universal binary)" to get the NBC assembler. This will get you the file nbc-1.2.1.r3.osx.tgz. Don't be alarmed by the weird name — your Mac can deal with it. When you click on the download link, you'll get a window that suggests saving the file; go ahead and save it. When the download completes, double-click on the file in the Downloads window (I'm using Firefox; every browser has a downloads window named something like that). Now, create a folder under your Applications folder or wherever you put stuff like this. I called mine "NXC." (That isn't very original, but it IS very obvious.) In Finder, find where your browser downloads stuff (most likely in <login name>/Downloads). When you doubled-clicked on the downloaded funky file name, it created a directory in your download folder called nbc-1.2.1.r3.osx. Move this folder to your NXC folder (or whatever you decided to call it).

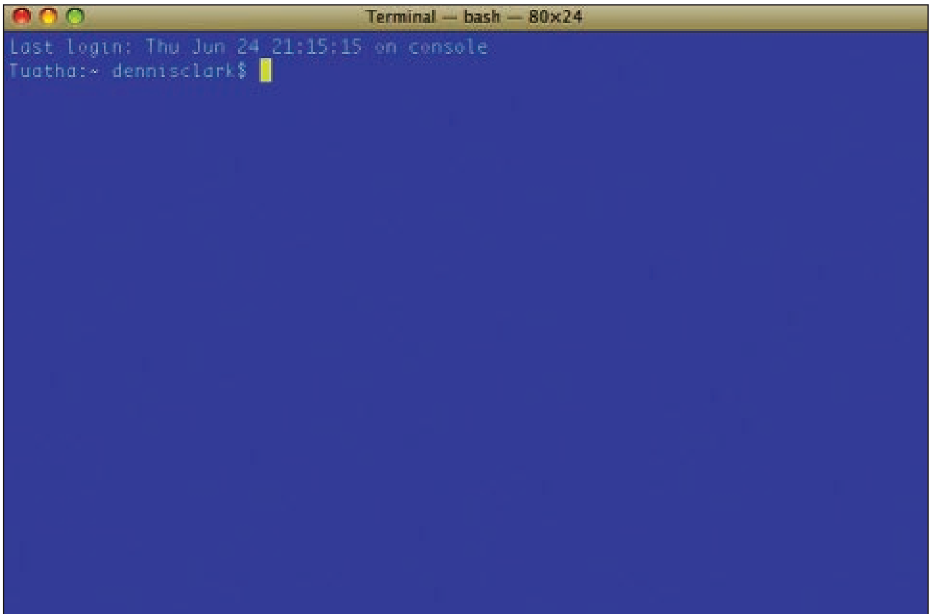


Figure 1. Terminal window.

Step Two: Do some UNIX command line stuff to install the tools.

Since you are going to be doing some command line work with Makefiles (don't panic) and stuff, you'll want to be able to work with the command line. It is now time to open a **Terminal Window**. You will find this application here: Applications/Utilities/Terminal.app. When you start this app, you get a terminal window with a command line that looks a bit like **Figure 1**. It won't look exactly like that because I've customized mine to a font, size, and color scheme that I like.

You will have to learn some UNIX commands. Fear not, they are few and simple for our purposes. The terminal will be set to your login directory which is just your login name (as mine is). To get back to this directory, type: **cd ~**. (The cd command means change directory in case you were wondering.) To find out where you currently are use the command **pwd** — which stands for Print Working Directory, again, if you are curious.

To see the files in this directory, type: **ls**. You'll see a bunch of directories. That's nice, but this isn't where we want to be; we want to be in the directory you created and moved the NXC folder to. To get there, type **cd /applications/nxc/nbc***. In OS X, you don't have to worry about letter case when just moving around, so I left everything lower case. The "*" at the end of that command is called a wildcard and we use it here so that we don't have to type out **nbc-1.2.1.r3.osx**. In this directory, you will find two files: one is nbc which is the NBC assembler; the other is called nxtcom_scripts.zip. You want to "unzip" this file to get the program that will allow you to communicate with the NXT brick. Type: **unzip nxtcom_scripts.zip**. You'll find lots more files in this directory now, including some other directories. (To tell the difference between files and directories, use **ls –F**. The –F will cause directories to have a trailing "/" and executable programs will have a trailing "*".)

Listing 1: Simple Makefile.

```
##################################################
# NBC Makefile for first.nxc
##################################################

## Project name and flags (if any)
PROJ = first

## Tool locations
NBC = /applications/nxc/nbc-1.2.1.r3.osx/nxt/nbc
NXTCOM = /applications/nxc/nbc-
1.2.1.r3.osx/nxt/nxtcom_scripts/nxtcom
INCS = /applications/nxc/nbc-1.2.1.r3.src/NXT

## Default stuff to make
all: rxe install

## Compile
rxe:
       $(NBC) -v=105 -Z1 -I=$(INCS) -
O=$(PROJ).rxe -nbc=$(PROJ).nbc -w+ $(PROJ).nxc

## Upload to device
install:
       $(NXTCOM) $(PROJ).rxe

## Clean target
clean:
       rm -rf $(PROJ).rxe $(PROJ).nbc
```

Listing 2: SONAR program.

```
#include "NXCDefs.h"
// Very simple implementation, everything is
placed in the main task.
task main(){
     short ultra;
     SetSensorLowspeed(S4);

     while(true){
       ultra = SensorUS(S4);
       ClearScreen();

       TextOut(0, LCD_LINE4, "Range");
       NumOut(0, LCD_LINE5, ultra);

       Wait(500);
     }

}
```

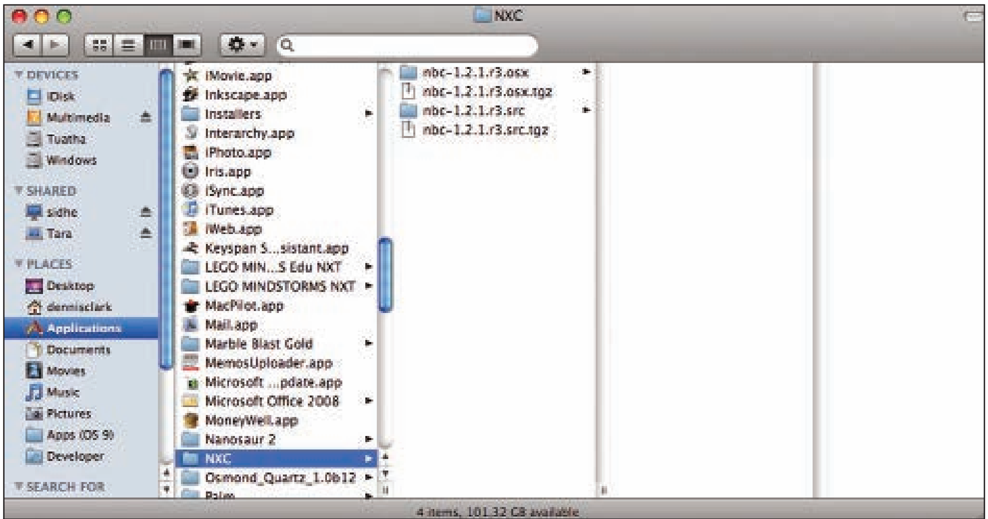


Figure 2. NXC application folder layout.

We're not out of the woods just yet. The nxtcom program cannot be executed; we have to change its file attributes. To make nxtcom executable, type **chmod +x nxtcom**. Now, we can assemble NBC files with the nbc program and download them to the NXT with nxtcom.

As I said before, I didn't know any of this when I started so I asked my friend Joe (thanks Joe!) to help me out. He sent me a simple Makefile that showed the steps to use to compile and download NXC code to the NXT brick. In **Listing 1**, you will see the Makefile which is a UNIX scripting language that "simplifies" using compilers, linkers, programmers, and other complicated command line things so you can type short and simple commands to get things done. (More on this later.) One key thing that you need to remember about a Makefile is that it needs to have the "M" at the beginning of the name capitalized. Why? Don't know, it's just an ancient UNIX custom.

It would take a whole column to explain this Makefile, so I won't do that. I'll just point out that it sets up where the compiler and download program are (nbc and nxtcom), and the compiler options to use to compile your code and download it over the USB port. Using the Bluetooth port can be set too. If anyone is interested, I'll do that question separately since it's more fun setting up the Bluetooth connections.

Listing 1 shows Joe's Makefile after I modified it. We'll come back to it later to explain what is going on and why I made the changes that I did. Joe initially put all of his source files in the same directory as the nbc assembler. I like to put applications in the Applications folder and programming projects in the Documents folder, so before we go any further, let's create a NXC projects folder here: Documents/NXC/projects/first.

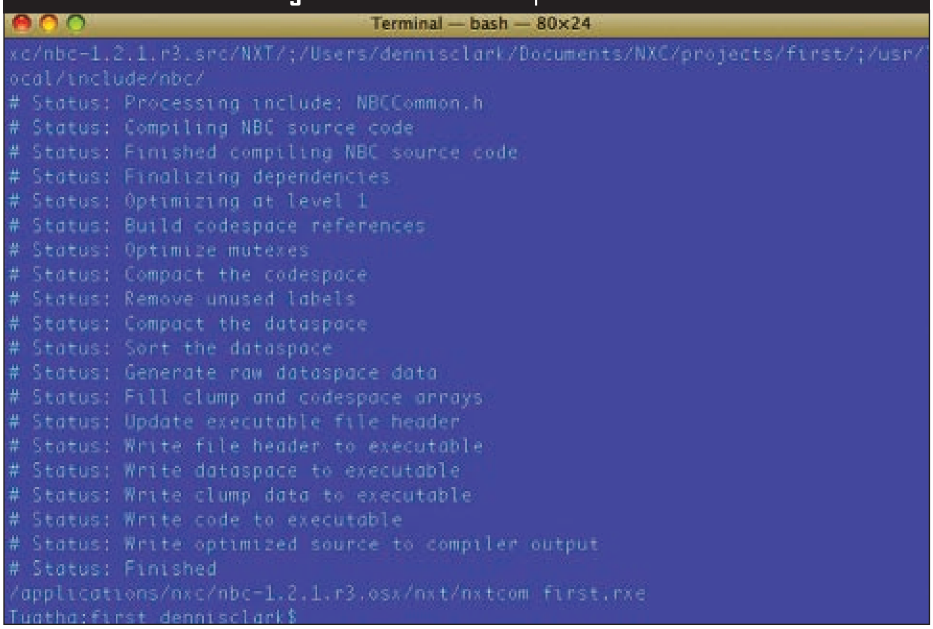
You can create this folder path in Finder or you can do it in your terminal window if you know the UNIX commands to do so (mkdir). (I did mine in Finder.)

Next, we'll create a simple program to read a SONAR sensor and display the results on the NXT screen. **Listing 2** shows this source. This was my first program written in NXC. These files are in the **first** folder we created in the last paragraph.

Let's see ... we've installed the NXC compiler, have a Makefile, and a source file — we're ready to go. I tried to make the program and it failed. The compiler could not find the NXCDefs.h file. I searched the SourceForge site and Googled everything that I could to find out where to get this file. I found no mention of it. When you take the "road less traveled" (Mac) and are using open source tools, you must be ready to dust off your brain cells to figure out how to get things to work. Many open source tools are built for the fun of it and out of the kindness of some hard working person. Usually, there are holes in the documentation that we must fill in while actually figuring out how things work. This was one such time.

After my fruitless search, I realized that NXC is a turn-key solution for Bricx Command Center. For the command

Figure 3. Successful compiler results.



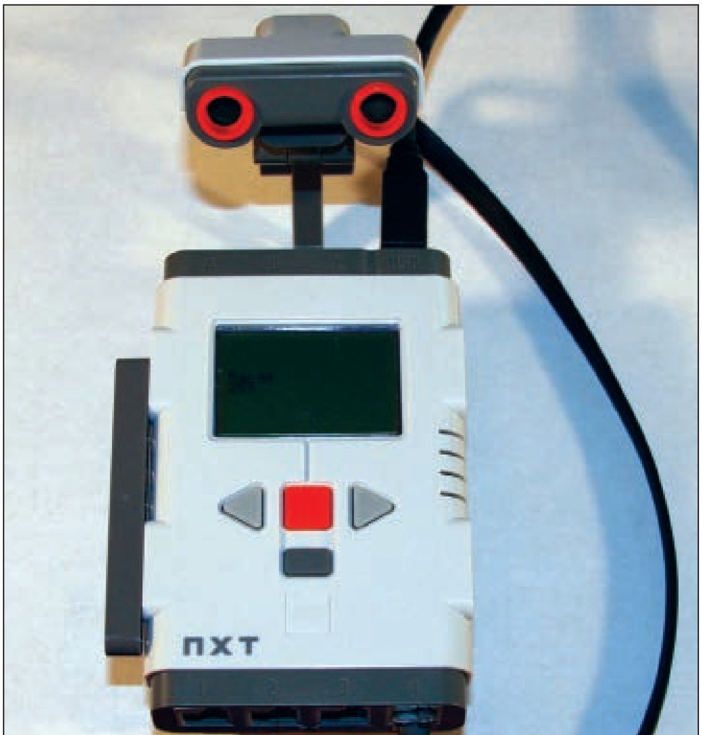


Figure 4. The NXT brick and SONAR sensor.

line, I needed to set up my own system. I then looked back at **http://bricxcc.sourceforge.net/nbc** and saw that there was a source release to go along with the executables. Hmm ... this sounds promising!

Step Three: Get the NXC/NBC source code too!

Using your favorite browser, download the source code by clicking on the link *NBC/NXC 1.2.1.r3 Source release*. Again, you will get a weird file that looks like nbc-1.2.1.r3.src.tgz. Your browser will ask you to save or open it with an archive utility. I like to save the files and open them later; your mileage may vary. After the file downloads, double-click on it so it will unpack. Using Finder, move the nbc-1.2.1.r3.src folder to your applications/NXC folder. In **Figure 2**, you'll see how I created these two essential folder locations.

Figure 5. NXT display output.

Step Four: Set up your Makefile, write your source code, then build!

If we look in the source folder in the NXC folder, we find — *ta da!* — our missing NXCDefs.h file! Now that we know where everything is, we have to *use* it. Believe it or not, that Makefile will make this a lot easier. Let's go back and look at **Listing 1** again. In this Makefile, I set up three specific locations: where nbc is, where nxtcom is, and where to look for the NXCDefs.h file.

```
## Tool locations
NBC = /applications/nxc/nbc-1.2.1.r3.osx/nxt/nbc
NXTCOM = /applications/nxc/nbc-1.2.1.r3.osx/nxt/
nxtcom_scripts/nxtcom
INCS = /applications/nxc/nbc-1.2.1.r3.src/NXT
```

If you set your folders up like I did, this Makefile should work just fine. Before you build your own program using the Makefile, you need to have your NXT brick connected via the USB cable and turned on. If you aren't, you'll get: **Error: could not open USB connection NXT**. If you have any program running on the NXT, you'll get: **Error: NXT Error #8f**. (You have been warned!)

To build your NXC program, all you need to do is be in your source code directory in Terminal. To get there from anywhere, type: **cd ~/Documents/NXC/projects/first**. (This is assuming you've created your source code folders as detailed above.) If you decide to use a different location, adjust that line accordingly. From your source code directory, type: **make**. That's it. You will see lots and lots of compiler output spew across your screen telling you everything that the compiler is doing.

The very last thing the Makefile does is download your code to the NXT brick. **Figure 3** shows the final outcome of the make command in the terminal window.

No project is complete without pictures, so **Figure 4** shows how I set up the NXT and SONAR sensor, and **Figure 5** shows what you should see on the NXT display when the program is running. I'm assuming that you have already used your NXT robot brick and know how to run a program. Using NXC to create and nxtcom to download the program puts the program on the NXT brick in the same way that the Mindstorms graphical programming language does. What you will notice is that NXC programs are much smaller than the Mindstorms programs. This is one of the advantages of using NXC.

To customize this Makefile for any program, all you need to change is the project name which is set here:

```
## Project name and flags (if any)
PROJ = first
```

I've included both my simple SONAR program and the Makefile in a zip file you can find at **www.servomagazine.com** under Mr. Roboto as *RobotoNXC.zip*.

Final Notes

We've just touched the surface of what you can do with NXC. For instance, there is a way to program and "talk" to the NXT brick via Bluetooth. If there's enough interest, I can do an article on that. There are also other tools that are more graphically oriented for the Mac that present some of the functionality that Windows users get with Bricx Command Center. You can find them at: **http://bricxcc.sourceforge.net/utilities.html**. I haven't investigated these yet, but it looks like there's a lot to play with (see **Figure 6**).

Figure 6. More tools (toys) for Mac NXT programmers.

Thanks again to those selfless individuals that toil away on open source. Those of us who aren't as adept definitely reap the benefits of your largesse!

That wraps up another Mr. Roboto column. As usual, if you have any robotics questions that you would like me to help you figure out, don't hesitate to drop me an email at roboto@servomagazine.com. I'll be happy to work on it! Until next time, keep on building those robots! **SV**

EVENTS Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**

— **R. Steven Rainwater**

SEPTEMBER

3-6 **DragonCon Robot Battles**
Atlanta, GA
In this competition, autonomous and remote control robots battle at the popular science fiction convention.
www.dragoncon.org

5-12 **Microtransat Challenge**
County Kerry, Ireland
Autonomous sail boats start on a race across the Atlantic ocean; this will take three to four months to complete.
www.microtransat.org

6-11 **National Junior Robotics Competition**
Science Centre, Singapore
100s of student teams compete in this LEGO Mindstorms competition.
www.science.edu.sg/ssc/events.jsp?type=17&root=0&parent=0&cat=317

15-19 **FIRA Robot World Cup**
Bangalore, India
This is the big annual robot soccer competition which includes RoboSot robot soccer, MiroSot micro robot soccer, HuroCup single humanoid robot soccer, AndroSot multiple humanoid robot soccer, and AMiRESot mini robot soccer. If your robot likes to play soccer, this is the place to be. Don't forget your vuvuzela!
www.fira.in

18 **RobotCup Junior Australia**
Canberra, Australia
In this competition, students and mentors create robots that compete in challenges such as robot soccer, robot rescue, and robot dance.
www.robocupjunior.org.au

18 **Robotour**
Bratislava, Slovakia
Autonomous robots must perform a navigation task in a park.
http://robotika.cz

OCTOBER

3-10 **Devyanin Mobile Robots Festival**
Moscow State University, Moscow, Russian Federation
Autonomous robot race.
www.mobilerobots.msu.ru/en

16 **Antimov**
SparkFun, Boulder, CO
This competition is autonomous robot performance art in which robots attempt to self destruct in humorous or entertaining ways.
www.sparkfun.com/commerce/product_info.php?products_id=9975

22-24 **Critter Crunch**
Denver, CO
Autonomous and remote control robots participate in the original robot combat event.
www.milehicon.org

22-24 **SRS Robothon**
Seattle, WA
This year's Robothon includes Robo-Magellan, Robot Combat, Double-Cross, and Brickheap Wars.
www.robothon.org

NOVEMBER

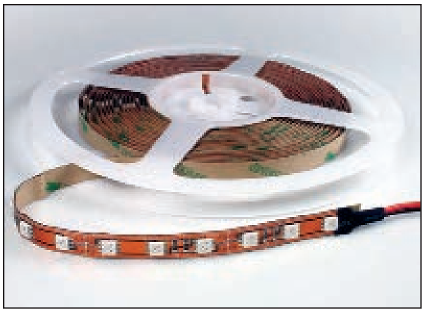
2-3 **Junior Robotics Challenge**
Singapore
This competition includes a line-following can collection.
http://jrc2009.webs.com

7 **International Micro Robot Maze Contest**
Nagoya University, Japan
This competition includes events such as Micro Robot Racer (1 cm cube), Climbing Competition (1 cm cube), Maze Solver (1 inch cube), and Two Leg Robot Competition (2 inch).
http://imd.eng.kagawa-u.ac.jp/maze

NEW PRODUCTS

ILLUMINATION

High Power Flexible LED Light Strips



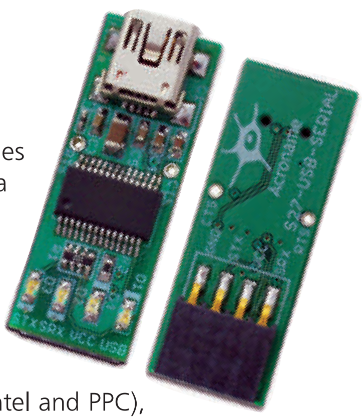
The NFLS-300X3 series strips from **superbright leds.com** use high power three-chip 5050SMD LEDs and are great for accent and other special application lighting designs. Available in cool, warm, and natural white color temperatures, these 12 VDC strips produce up to 350 lumens per foot while drawing just 0.3 amps per foot. Other colors available include blue, green, red, and amber. They can be cut in three-LED (2 in) segments and have an adhesive backing to easily handle contoured installations.

For further information, please contact:

SuperBrightLEDs Website: **www.superbrightleds.com**

SERIAL ADAPTERS

Serial Adapter



This serial adapter from Acroname Robotics bridges UART serial data to USB as a virtual serial port. One can communicate with a device that requires serial data communication. Driver support for this device is complete across Windows, Mac OS X (including both Intel and PPC), Linux, and many other operating systems.

This connector conforms to the Acroname Serial Interface Standard allowing it to be used with several different devices. The TTL side is also 3.3V tolerant if you are using lower voltage circuits than 5V. Power for the interface bridge connector circuitry is taken from the USB bus. Four LEDs on the adapter help troubleshoot connections between a host computer and a device. Serial data being transmitted to a device is indicated by the STX connection with a green LED. Serial data being received from a device is indicated by the SRX connection with a blue LED. A VCC status that may be used to show whether a device is powered on is indicated with a red LED. USB status connection is shown with a yellow LED. Price is listed at $12.50.

For further information, please contact:

Acroname Robotics Website: **www.acroname.com**

INDUSTRIAL CONTROLLERS

Open Architecture Workcell Controller

EPSON Robots introduces its new RC620+, EPSON's fourth generation open architecture PC-based workcell controller. The high performance RC620+ provides ease of use with EPSON RC+ development software. The RC620+ is capable of controlling up to four SCARA robots, three six-axis robots, or two SCARA and two six-axis robots for a total of up to 20 axes of motion from one controller.

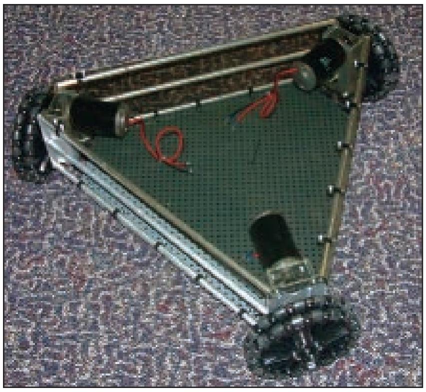
Three PCI slots are available for options such as vision systems, fieldbus masters, or third party plug-in cards to expand the RC620+ controller as needed for particular applications. The RC620+ runs a separate CPU dedicated to motion control and real time options including additional I/O, conveyor tracking, expansion I/O, and DeviceNet, Profibus, and EtherNet/IP fieldbus slave options. In addition to these features, the RC620+ controller also provides RC+ Controls software. At the heart of it is EPSON's Real Time Engine — capable of controlling up to 48 simultaneous tasks. There's also the SPEL+ Language, and motion and I/O control.

For further information, please contact:

EPSON Robots
18300 Central Ave.
Carson, CA 90746
Tel: **562•290•5910**
Website: **www.robots.epson.com**

PLATFORMS

Drive Platform



The new Tri-Lamda drive platform from AndyMark is a purely omni-directional drive system kit that is easy to build and dependable. It is shipped as a kit of parts which includes:

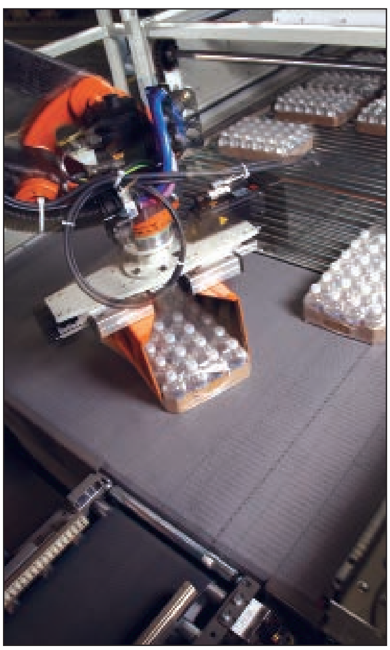
- Toughbox Nano (3)
- CIM Motor (3)
- Encoder Package (3)
- 500 Key Hub (3)
- 8" Dualie Plastic Omni Wheel (3)
- Bottom perforated sheet
- Top smoked plastic sheet
- Edge brackets for Toughbox Nano (6)
- Side tubes, aluminum (9)
- Associated fasteners cost is $745, and CAD file and layout options are coming soon.

For further information, pleast contact:

AndyMark, Inc. Website: **www.andymark.biz**

MATERIAL HANDLING

Robotic Arm Retrofits to Existing Alvey In-Line Palletizers



Intelligrated®, announces the launch of the Alvey® Robotic Retrofit Program for its family of Alvey in-line case palletizers. The program includes installation of one or more jointed-arm robots to create a hybrid palletizer that combines robotic flexibility and repeatability with the speed and reliability of conventional palletizer layer deposition. Benefits of the Alvey retrofits include gentler positioning and rotation of product, automatic line changeovers, pattern changing flexibility, and extended operational life at a lower cost than a new robotic palletizing solution.

Retrofits are completed on-site by removing the slat divider and existing case turners from the Alvey in-line case palletizer and installing one or two robotic arms for pattern forming. Intelligrated offers each customer a choice of robotic arm OEM and operating software. Every robotic retrofit includes design and manufacturing of end-of-arm tooling, OIT (Operator Interface Terminal) screens, infeed conveyor, and PLC programming.

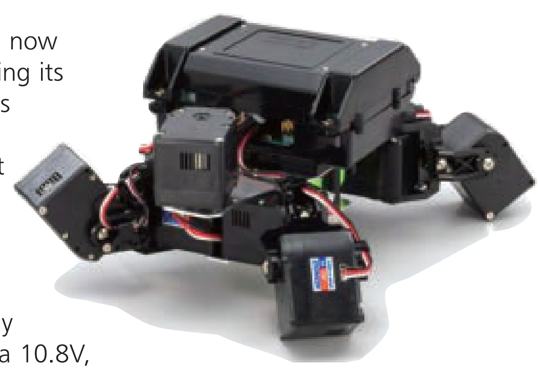
Flexible pattern forming from the OIT allows operators to handle changing packages and multiple SKUs quickly and easily. Patterns can be programmed to orient high-graphic packaging, creating ready-to-sell pallets quickly and easily. With quick product reconfiguration and instant line changeovers, the Alvey Robotic Retrofit achieves gentle packaging rates in excess of 100 cases per minute.

For further information, please contact:

Intelligrated Website: **www.intelligrated.com**

ROBOT KITS

Turtle Robot Kit



Kondo is now producing its animal series with the first product being a turtle. It's a simple kit with nine servos, a tiny board, and a 10.8V, 300 mAh NiMH battery pack, complete with software and a frame.

The new robot has four legs — each with two servos — plus a pannable head, adding up to a total of nine degrees of freedom. The servos are Kondo KRS-4024SHV, and the robot controller is the RCB-3HV. As much as possible, the kit uses standard KHR robot parts, though the base mounting bracket appears to be a new design.

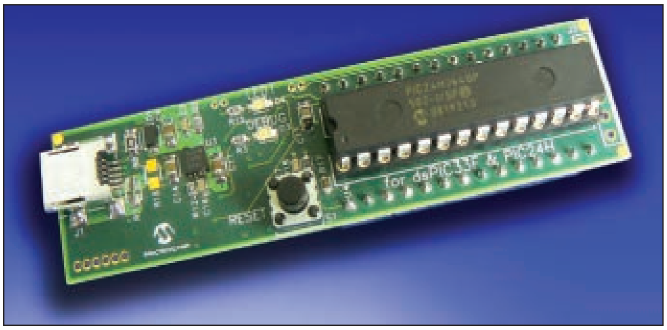
Kondo recommends options like spare batteries, a remote control, and even a screw driver set.

List price in Japan for the kit is 39,900 yen, or about $450 US dollars.

For further information, please contact:

Kondo Website: **www.kondo-robot.com**

DEVELOPMENT BOARDS



Microstick Development Board

Microchip Technology, Inc., announces the Microstick for dsPIC33F and PIC24H development board which provide a complete, low-cost solution for designing with Microchip's 16-bit PIC24H microcontrollers and dsPIC33F Digital Signal Controllers (DSCs) in a compact 20x76 mm footprint. At a cost of $24.99, the Microstick offers an integrated USB programmer/debugger which shortens learning curves. For maximum flexibility, the Microstick can be used stand-alone or be plugged into a prototyping board. Educators are eligible for a 25% discount.

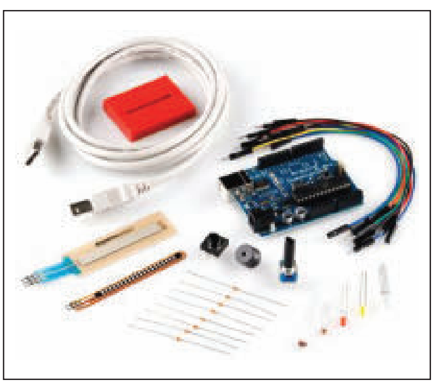
The Microstick is populated with a socketed microcontroller that can be easily swapped out. It works with the PIC24HJ64GP502 and the dsPIC33FJ64MC802 DSC. Software support includes the same free MPLAB® Integrated Development Environment (IDE) and software libraries that work with all of Microchip's 8/16/32-bit PIC® microcontrollers and DSCs. Additionally, the dsPIC33F DSCs are supported by a free demo version of Microchip's Device Blocksets for the MATLAB® language and Simulink® environment.

For further information, pleast contact:

Microchip Technology, Inc. Website: **www.microchip.com**

MICROCONTROLLERS

Arduino Starter Kit



An Arduino Starter Kit now available from Micro Controller Pros Corp. includes a multitude of inputs, outputs, and sensors to help get you started in the world of Arduino.

The Starter Kit includes:
- *Arduino Duemilanove ATmega328* — the latest Arduino USB board, fully assembled and tested.
- *6' USB A to B cable* — USB provides power for up to 500 mA (enough for most projects) and is ample length to connect to a desktop or laptop USB port.
- *Miniature breadboard* — Useful for making circuits and connections off the Arduino. Breadboard may come in various colors.
- *Male to male jumper wires* — High quality wires that allow connection to the female headers on the Arduino to the components and breadboard.
- *Flex sensor* — A 4.5 inch sensor from Spectra Symbol that is flexed; resistance increases from ~900 ohms straight to ~22,000 ohms at 180 degrees.
- *SoftPot* — By pressing down on various parts of the 50 mm strip, the resistance changes linearly from 100 ohms to 10,000 ohms, allowing accurate calculation of the relative position on the strip.
- *Photocell* — A sensor to detect ambient light.
- *Thermistor* — A sensor for detecting ambient temperature and temperature changes.
- *Tri-color LED* — Can be used to PWM mix any colors needed.
- *Basic LEDs* — Light-emitting diodes for use as indicators.
- *Linear trim pot* — Also known as a variable resistor, this is a device commonly used to control volume, contrast, and as a general user-control input.
- *Buzzer* — For making a range of noises, alarms, and possibly music.
- *12 mm button* — Larger size for ease of use.
- *330 ohm resistors* — Five current-limiting resistors for LEDs, and strong pull-up resistors.
- *10K ohm resistor* — Can work as a pull-up, pull-down, or current limiter.

Price of the kit is $59.95.

For further information, please contact:

MicroController Pros Corporation Website: **http://microcontrollershop.com**

RFID

Serial RFID Read/Write Module

Designed in cooperation with Grand Idea Studio (**www.grandideastudio.com**), the Parallax Radio Frequency Identification (RFID) Read/Write module provides a low-cost solution to read and write passive RFID transponder tags up to three inches away. The RFID transponder tags provide a unique serial number and can store up to 116 bytes of user data which can be password protected to allow only authorized access.

The RFID module can be used in a wide variety of hobbyist and commercial applications, including access control, user identification, robotics navigation, inventory tracking, payment systems, car immobilization, and manufacturing automation.

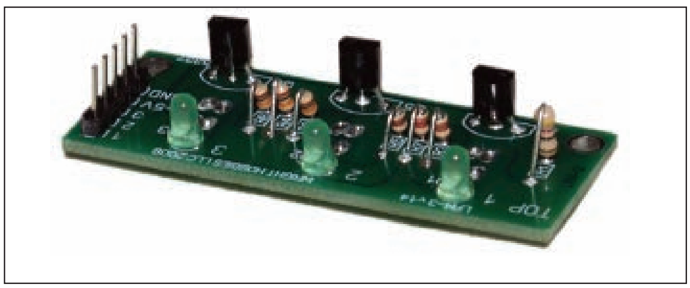
Other features include an optional security feature to prevent the tag from being read or written to without a password and a bi-color LED for visual indication of status.

For further information, please contact:

Parallax, Inc.	Website: **www.parallax.com**

SENSORS

Line Following Module



The LFM-3 from Wright Hobbies Robotics is a sensor module that can detect a white line on a black surface or a black line on a white surface. The LEDs will light when the sensors detect a white or reflective surface. The output from each sensor is a variable voltage of ~0V for black, non-reflective surfaces to ~4.5 to 4.9V for white or reflective surfaces. The voltage swing is sufficient to be read as a digital input, as well as an analog voltage.

The Sharp GP2S40 reflective surface sensors use infrared light to detect a white or reflective surface at a range of about 4 mm-8 mm. The sensor module can detect lines at larger distances, up to about 15 mm.

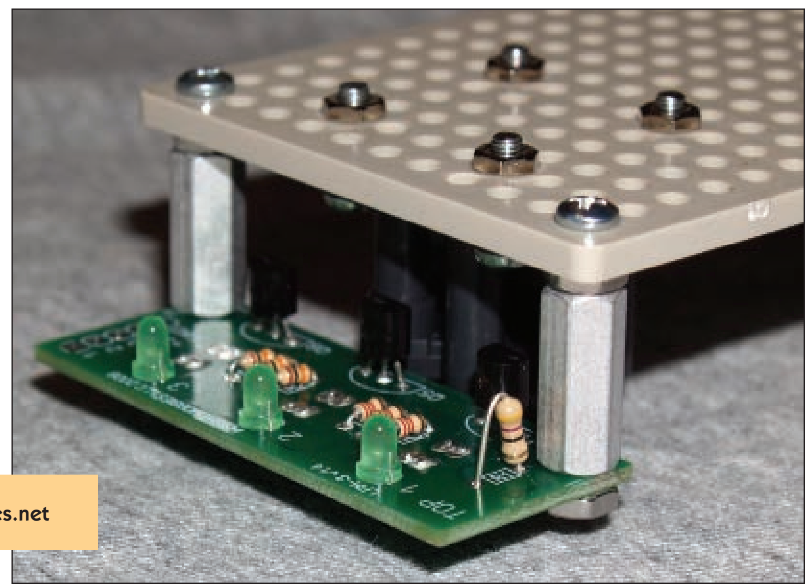
You can adjust the height above the surface according to your needs.

The LFM-3 is designed to allow a robot to follow a 3/4" wide line. The module is all through-hole components, making it simple to assemble.

The GP2S40 sensors use infrared light to detect surfaces near it. Three sensors provide the ability to accurately detect and follow a line.

Price is currently $14.95.

For further information, please contact:



Wright Hobbies Robotics	Website: **www.wrighthobbies.net**

bots IN BRIEF

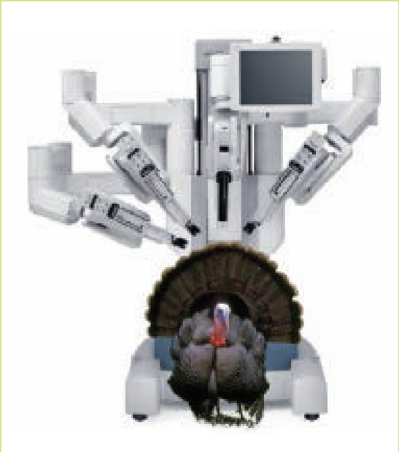
HUMAN PURRSONALITY

It's definitely true that human facial features are great at communicating emotional information, so there's certainly something to be said for incorporating things like eyes and eyebrows into a robot. Flobi — a robot from Bielefeld University — is a good example of a conscious decision to make a humanoid robot head that is capable of recognizable human expressions while not being totally creepy.

Flobi relies on expressive elements that are almost cartoonish in their simplicity; 18 actuators move the eyes, eyelids, eyebrows, and mouth, and there are LEDs in Flobi's cheeks so it blushes … 'it' because Flobi's hair (all of the facial elements, in fact) can be easily changed to look male or female.

SWEET DEAL

This robot cookie jar was made by Vandor and is based on some of Japan's '50s and '60s designs. The ceramic collectible stands about 8" tall and is supposedly no longer being made. What's left is available at Amazon.



FOR THE BIRDS

More and more, people are starting to specifically request robot-assisted surgeries as opposed to having just a human operate on them. Now, researchers at Duke are working on an entirely autonomous robot arm that can take biopsies on humans based on ultrasound data. It works pretty well, at least so far on the dead turkeys that they've been practicing on.

"In the latest series of experiments, the robot guided the plunger to eight different locations on the simulated prostate tissue in 93 percent of its attempts."

Hmmm ... wonder about that other 7 percent … At least the robot didn't go berserk and wildly stab everything within reach.

In any case, the idea is that eventually (soon, perhaps?) robots will be able to (at the very least) take care of simple, routine medical procedures which will save patients both time and money.

Turkeys are used because they have similar flesh to humans, and they show up about the same on an ultrasound.

Neither this daVinci sytem nor turkey were actually used or harmed in this study.

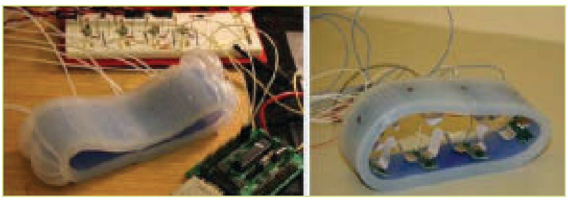
bots IN BRIEF

(NOT) IN THE DRIVER'S SEAT

A pair of robotic vehicles from Vislab (the artificial vision and intelligent systems lab at the University of Parma) departed Parma, Italy in July for Shanghai, China. The 100% electric vans will travel 8,000 miles over three months, enduring (hopefully) all kinds of extremes ranging from downtown Moscow to the Gobi desert.

These vans are actually "autonomously" following a vehicle that's being driven by a human. The vans have been kitted out with the same sort of obstacle detection and avoidance tech as the DARPA Grand and Urban challenge vehicles.

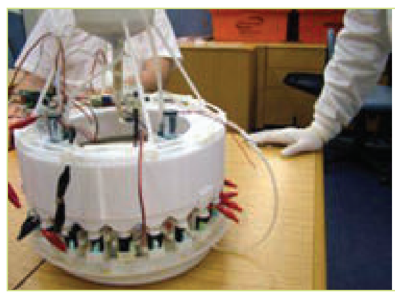
For now, this technology is targeted mostly at goods transport as opposed to letting you take a nap while your car drives you somewhere.



ON A ROLL

This latest DARPA Chembot prototype is a collaboration between iRobot, MIT, and Harvard, and like its Chembot compatriots, the system of movement it employs is quite simple: by selectively inflating compartments on its exterior, the robot can roll itself forward. Ultimately, this robot would be powered by chemical pressure (instead of compressed air) achieved through thermal expansion or phase transition or "smart fluid" (whatever that is).

NOW, THIS IS THE SH...

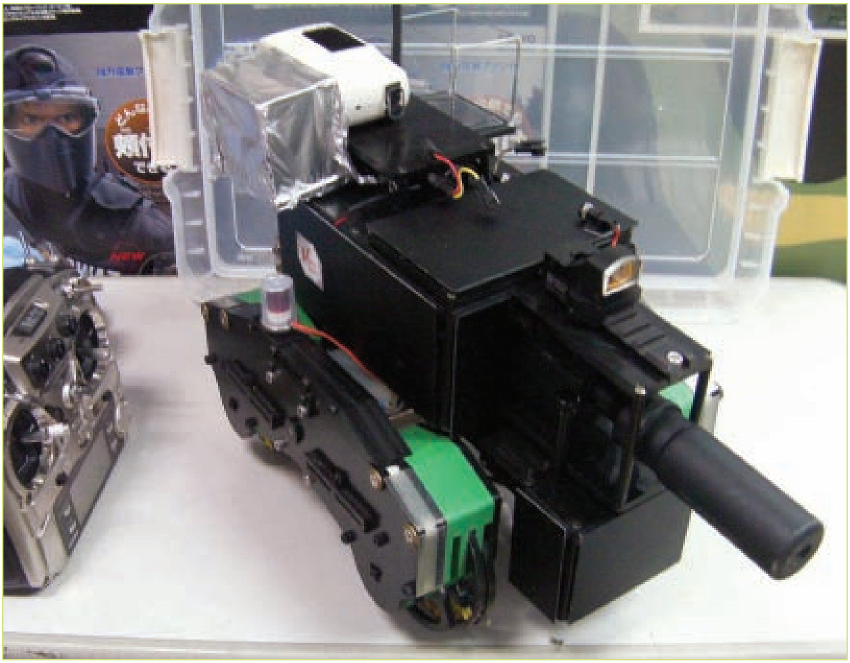


This robot goes poo. Ecobot III contains a fully functional digestive system capable of ingesting biomass, turning it into energy, and then excreting waste. Lovely. The actual digestion process is done by a series of microbial fuel cells (MFCs), where bacteria chow down and produce hydrogen atoms as a byproduct. The hydrogen goes into a fuel cell which generates electricity to power the robot, plus pure water which the robot then drinks to keep itself from getting dehydrated. The remaining biomass goes through the entire cycle once more, and then it's, um, purged.

Director of Bristol Robotics Laboratory, Chris Melhuish, said MFCs had been tried before but an artificial gut was needed to solve the problem of previous models which was that humans had to clean up the waste left by bacterial digestion. Melhuish explained the robot was called Ecobot III, but admitted "diarrhea-bot would be more appropriate, as it's not exactly knocking out rabbit pellets."

The difference between Ecobot and other robots that use biomass for fuel (like EATR) is that Ecobot digests things to produce energy rather than burning them to generate heat to boil water to create steam to produce energy. Thanks to its bellyfull of microbes, Ecobot is actually able to digest things, and this makes it much more adaptable when it comes to sources of fuel, since it's able to run on stuff that doesn't burn (like waste water). Yep, this robot not only poos, it can potentially be powered by poo. At the moment, Ecobot III is only 1% efficient, and while it's technically capable of operating for several days completely on its own, it can't really do much in that time.

Cool tidbits herein provided by Evan Ackerman at www.botjunkie.com, www.robotsnob.com, www.plasticpals.com, and other places.



SURVIVAL OF THE 'HIT'TEST

The long awaited Robot Survival Game took place in Japan in July with eight robots entering the fray. The event was similar to Mech Warfare in that the robots were armed (with airsoft guns) and controlled remotely via onboard cameras. The most significant difference between the two competitions seems to be that the Survival Game doesn't penalize robots for having extra legs and allows other mobility systems — leading to some pretty interesting designs.

The hit sensors are actually quite brilliant in their simplicity and effectiveness. The 'tinfoil hats' cover light sensors, and when holes are shot in the tinfoil, the sensors register more light. When a certain threshold is reached, the robot is declared destroyed.

Main regulations for the Robot Survival Game included:

1) Mobile bodies are not only multi-legged robots or biped robots, but also can be mobile by means of wheels or caterpillar style traction.
2) Weapons are toy guns under 0.1 J and/or other similar shooting systems.
3) The equipped stopping mechanism of the body is by gun-hit sensor using a solar battery covered with tinfoil (6 x 6 x 6 cm cubed or 5 x 6 x 7 cm).
4) Robot control has to be done via images on a PC from a camera attached to the robot.
5) Participants have to handle toy guns carefully and put eye protectors on.

GETTING HAWKED

Having proven itself with over 30,000 combat hours performing surveillance, Northrop Grumman's Global Hawk UAV is heading to Europe as part of a joint venture with EADS (the European Aeronautic Defence and Space Company). Renamed the Euro Hawk, the UAV is getting a brand new surveillance system to accompany its already impressive airframe capabilities which include a 15,000 mile range and 36 hours of endurance at over 60,000 feet.

EADS is also working on a number of other UAV and UCAV projects, although budgetary concerns have put a bit of a damper on things.

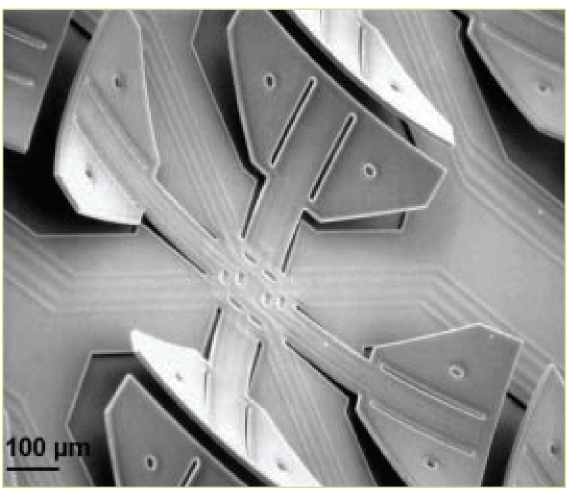
IT WAS AN ITSY, BITSY, TEENIE, WEENIE ...

This microbot — born out of a collaboration between the University of Washington and Stanford — weighs half a gram, is about the size of a dime, and is thinner than your fingernail. This robot actually started out as a prototype part for a paper-thin scanner or printer back in the '90s. Then, the same hardware was used to design a docking system for picosatellites. It was only recently that researchers flipped the thing over on its back and discovered that it could not only walk, but carry up to seven times its own weight while moving in any direction at a blistering three feet per hour.

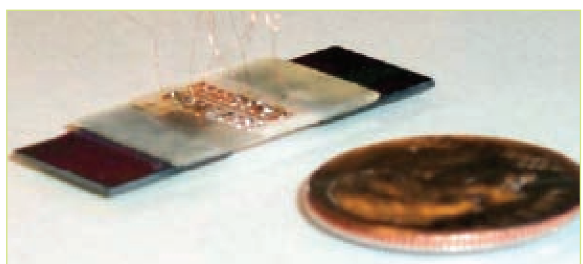
The microbot has 512 teeny tiny individual legs, grouped into 128 clusters of four, positioned orthogonally to one another to allow for movement in any direction. The legs consist of an electrical wire sandwiched between two different thermally reactive materials. When electricity runs through the wire, the materials heat and the outside one expands, forcing the foot to curl. When the current is turned off, the foot cools, returning to its original position. At such small scales, this process can happen very rapidly, and the feet are able to complete a movement cycle between 20 and 30 times every second.

Graduate students added paper clips to the microrobot's back to test how much weight it could carry. The robot could carry seven times its own weight.

Since the microbot didn't start off life as a robot, it's not exactly optimized. For example, minor modifications could likely reduce its weight by about 90% which would boost its power-to-weight ratio (and potential payload) even higher. This means it should be relatively straightforward to equip the robot with a battery, circuit board, and sensors, making it entirely autonomous.



Tiny, four-sided cilia — pulsating structures that mimic the hairs that line the human windpipe — are arranged in rows along the underside of the robot.



The microrobot is about the width of a fingernail, significantly slimmer than a dime. Wires to the center transmit power and directions. At the front and back is an 8x8 grid of tiny shuffling legs.

ARMED AND (REMOTELY) DANGEROUS

Recently, South Korea deployed an armed surveillance robot at a guard post within the demilitarized zone in Gangwon Province. The robot consists of an array of cameras and sensors plus a 40 mm grenade launcher.

The robot is remote controlled and it appears to have autonomous sensing and targeting capability, but it doesn't fire autonomously. Rather, it's fired remotely by a human. In this respect, it's much the same as a Predator drone or a Talon SWORDS, except that it's stationary.

This approach can be very efficient if you have a lot of static area to monitor, since several robots can be controlled by a small group of humans — with each robot only alerting its controllers if its sensors detect something relevant. Each of the robots cost $330,000.

If this trial proves successful, South Korea plans to deploy the robots at all guard posts along the DMZ, and possibly to some offshore islands.

GOLD FINGER, ER, UH, ARM

MAKO Surgical Corp's RIO® Robotic Arm Interactive Orthopedic System has received a 2010 Gold Medical Design Excellence Award. RIO is used to perform MAKOplasty® partial knee resurfacing — a treatment option for patients with early to mid-stage osteoarthritis of the knee.

"It is an honor to receive this recognition," said Dr. Maurice R. Ferre, President and Chief Executive Officer of MAKO. "RIO, which was released last year, represents an important expansion to the previous generation of our robotic arm surgical system, incorporating enhancements such as improved dexterity and a more ergonomic configuration. Additionally, the RIO is an enabling platform for current and future applications."

The Medical Design Excellence Awards competition (**www.MDEAwards.com**) is the only awards program that exclusively recognizes contributions and advances in the design of medical products. Entries are evaluated on the basis of their design and engineering features, including innovative use of materials, user-related functions that improve healthcare delivery and change traditional medical attitudes or practices, features that provide enhanced benefits to the patient, and the ability of the product development team to overcome design and engineering challenges so that the product meets its clinical objectives.

TEAMING UP FOR NOVEMBER COMPETITION

Six high-tech science and technology teams from four continents have been named finalists in the inaugural Multi Autonomous Ground-Robotic International Challenge (MAGIC 2010). Teams from the United States, Turkey, Japan, and Australia have been selected by the US and the Australian departments of defense to compete this November in Australia in an effort to develop the next generation of fully-autonomous ground robots.

The finalists who will compete at the Royal Showground in Adelaide, South Australia, November 8-13, are:

- Cappadocia - Ankara, Turkey
- Chiba - Tokyo
- Magician - Perth, Australia
- RASR - Gaithersburg, MD
- Team Michigan - Ann Arbor, MI
- University of Pennsylvania - Philadelphia

The competition's aim is to develop teams of robots which can operate autonomously on the battlefield in dangerous situations helping keep soldiers out of harm's way.

RACK & ROLL

Innovative lift truck applications involving unmanned operation are being increasingly explored for high-risk work environments, such as those in the military, according to Toyota Material Handling, U.S.A., Inc. This past June, the US Army Logistics Innovation Agency (LIA) hosted demonstrations at Fort Lee, VA, of an MIT-developed prototype unmanned robotic Toyota lift truck capable of locating, lifting, moving, and placing palletized supplies within an existing outdoor supply depot. The demonstration included review of the robot's safety features, sensor capabilities, and human-robot interface based on voice and gesture commands.

The robotics technology was developed by the Massachusetts Institute of Technology (MIT) BAE Systems and Lincoln Laboratory in collaboration with the LIA, the Combined Arms Support Command (CASCOM) Sustainment Battle Lab, the Office of the Secretary of Defense (OSD), and the Director of Defense Research and Engineering (DDR&E).

The 3,000-pound capacity, internal combustion Toyota 8-Series lift truck was modified by researchers at MIT's Computer Science and Artificial Intelligence Laboratory to perform embodied speech and gesture understanding; shape estimation (from laser range scanner data); machine vision (from camera data); motion estimation (from GPS, inertial data, and wheel odometry encoders); and autonomous mobility and pallet manipulation. Proprietary Controller Area Network (CAN-bus) protocols (provided by Toyota's 8-Series product engineering team) enabled the MIT team to connect its algorithms directly to the lift truck's manual and electrical controls.

NEW DISTRIBUTOR FOR COROWARES ...

CoroWare Technologies, a subsidiary of CoroWare, Inc., has entered into a distribution agreement with Singapore Robotic, a supplier of quality technology products.

Singapore Robotic offers a broad range of technology components and tools used in robotics development. The company's online store (**www.sgbotic.com**) includes products from advanced robotics systems and books, to SFE widgets and tools.

CoroWare's robotics product line includes the CoroBot and Explorer platforms. These robots include PC-class processors, on-board sensors, and color cameras. Options include laser range finders, pan-tilt cameras, and robotic arms. CoroBot is designed for development inside the lab. Explorer is built on a more rugged chassis to withstand certain environmental elements encountered in the field, such as dust, dirt, gravel, sand, leaf debris, and small puddles.

CoroBot

Explorer

COMBAT ZONE

Featured This Month:

Features

32 *PARTS IS PARTS: Let's Roll – Wheels for Combat Robots*
by James Baker

35 *MANUFACTURING: Creating Composite Combat Bots*
by Bradley Hanstad

38 *COMBAT ZONE'S GREATEST HITS*
by Kevin Berry

Events

38 *June/July 2010 Results and Sep/Oct 2010 Upcoming Events*

PARTS IS PARTS:

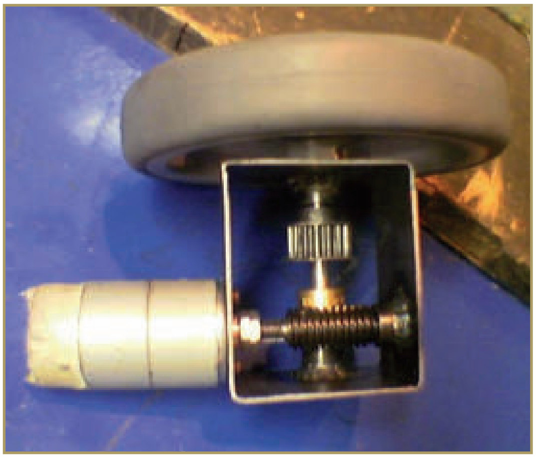
Let's Roll – Wheels For Combat Robots

● by James Baker

It does not matter how powerful your robot is, if you cannot transfer that power to the floor. Wheels are the single most important part of your robot because without them doing their job properly, the effectiveness of every other capability is reduced.

Power is nothing without the ability to use it, but to many, the only consideration is to have as much grip as possible from your wheels. Assuming you can get the grip you want, what consideration is actually given to the wheel structure itself?

You can buy wheels from hundreds of suppliers all over the world. They are usually round (but not always!) and come in many different sizes. It is not a simple task, however, to buy wheels that will work in robot combat. There are only a few people around the world who sell ready-made robot combat wheels, and of these suppliers, some may be using commercially available wheels that are modified or adapted for robot use. Or, they may offer custom machined wheels that they have manufactured themselves. I cannot offer you a guide to buying your wheels from one of these suppliers, as I have not bought from many of them



A commercially available wheel mounted to a live axle.

and so cannot attest to their quality and effectiveness. However, I can offer you a guide to making your own wheels — both by way of modifying commercially available wheels or machining your own.


Brittle plastic wheels will fail in robot combat.


Bespoke wheels with hubs for a dead axle.


These large titanium wheels had to be custom made.

Modified Wheels

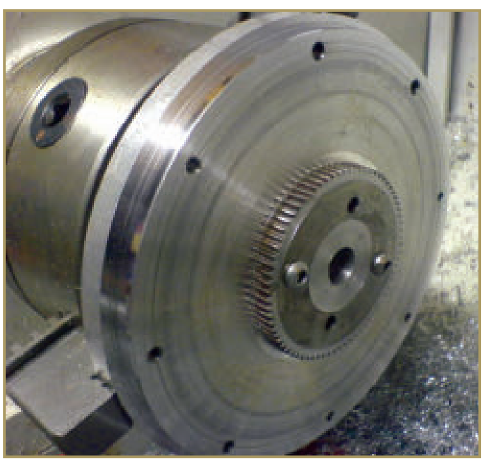
This is a really big subject. You can choose from so many sizes, types, and materials when you look for commercial wheels. From one inch nylon casters to 18 inch automotive alloy wheels, just about every possible wheel type has been tried on robots with varying levels of success. There are so many successful applications of commercially available wheels, I must generalize at this point to fit onto these few pages.

When choosing a wheel off the shelf, the first thing to consider is the diameter. Your robot size, shape, drive motor requirements, and gear ratios will give you an ideal theoretical wheel diameter, so aim to get as close to the ideal size as possible. There are many ways to work out the size of wheel you will need, but if you are not comfortable doing the mathematics needed to calculate the torque the robot needs to be effective and how to calculate the gear ratios needed for your motors to do this, I suggest you look to other robot builders and copy their solutions. You can then tweak performance by changing to the next size wheel for more speed or pushing power.

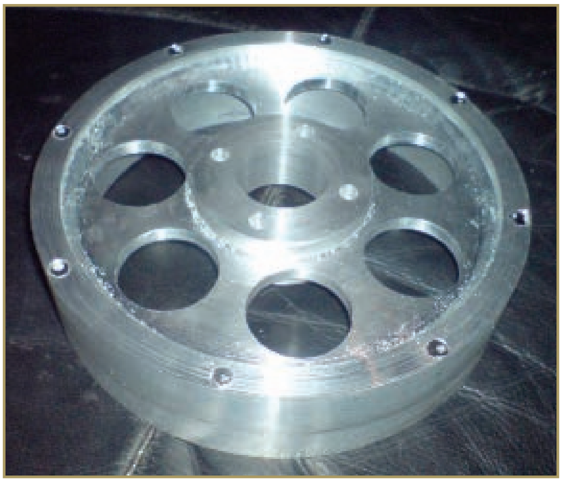
Material is the next issue. Commercial products are usually designed for a specific application. Some wheels may be plastic, pressed steel, or cast iron. A plastic wheel may be hard and brittle, or soft and flexible. It may deform over time due to the weight of the robot. It may warp from heat. The friction material bonding may degrade and the wheel may shed its tire. A metal wheel may be made of strong steel, fragile cast iron, or any number of different materials that offer advantages and disadvantages. As long as you do your research, you should find a suitable material.

The next factor to consider is whether you want to run with a live or dead axle. A live axle rotates with the wheel and the drive torque from the motor is transferred to the wheel directly through this axle. A dead axle is fixed to the chassis and has bearings mounted to it upon which the wheel rotates. Torque is then transferred to the wheel directly with the shaft taking no drive torque. The main physical difference as far as the wheel itself is concerned is that the wheel designed for dead axles will tend to have bearings or bushings fitted when you buy it; those for live axles have a plain bore or no bore.

Hub adapters are commercially available for use on live axle wheels and they all do the same basic job. They convert the output shaft from the gearbox to a size and shape that fits the wheel center. Some hub adapters convert from round shafts to hex shaft, or are keyed. Others convert the wheel bore to the shaft bore, and some have a thread cut into them (or appropriately sized nuts fixed into them) to screw onto


Machining billet aluminum to make a wheel.


Machining your own wheels can be very rewarding.



Drilling holes will reduce the overall weight even further.

a threaded output shaft. If you cannot find a suitable hub adapter, these are a lot easier to make than bespoke wheels, so machining your own hubs for your commercial wheels is often a good compromise.

The other factors to consider are the maximum safe RPM of the wheels, the maximum specified load, the friction (tire) material, and bonding methods. Trial and error plays a large role in the choice of commercial wheels.

Machining Your Own Wheels

With so many types of wheels available to purchase, why would anyone make their own? My robot 'Wheely Big Cheese' uses custom built wheels that are basically welded titanium 'drums' with domed front and rear faces, and a keyed aluminum hub for the titanium live axle. Instead of tires, the wheels have tungsten spikes that bite into wooden arena floors. The entire wheel weighs just 10 lbs and easily survives most combat situations. Nothing commercially available met my requirements for the same weight which is exactly one of the reasons to justify making your own wheels.

The first key factor when designing your own wheel is the weight. This can be approximated in advance by first deciding its dimensions. From these dimensions, you should be able to calculate the wheel volume. Then, by using the density information of various materials and multiplying this by your wheel volume, you can approximate the weight of the wheel for different materials. If the wheel is going to be too heavy, many things can be done to make it lighter, including manufacturing it from a lighter material.

The most obvious way of reducing the weight of a wheel of given diameter and material density is to reduce the only other variable: its width. This is a linear relationship, so in order to lose 50% of the weight, you simply cut the width by half. This, of course, reduces the grip available from the wheel, as the contact patch of the tire is now halved. So, other solutions that keep the same tire width may be more desirable.

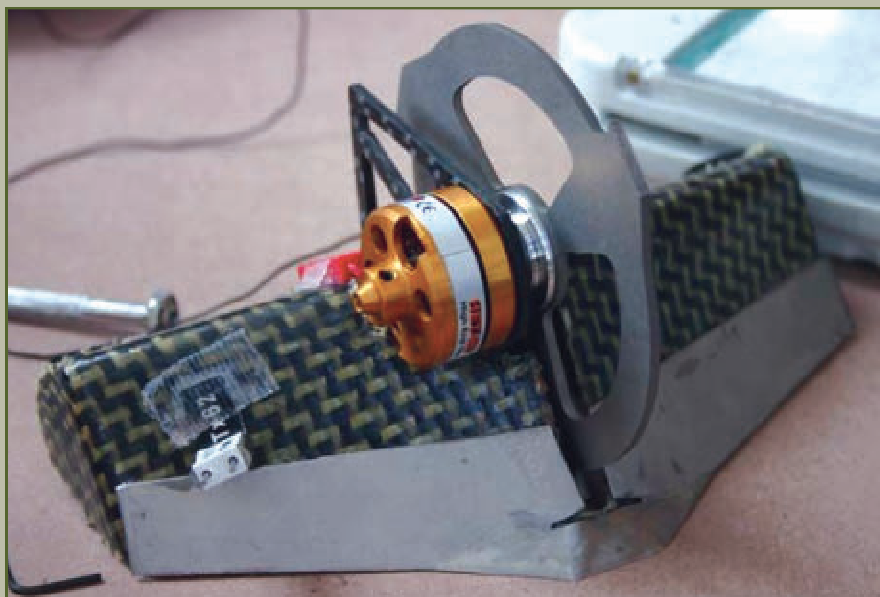
Machining out the material from between the outer and inner edges — and creating what is effectively an 'I beam' section across the wheel — can remove significant amounts of weight without compromising the strength of the wheel too much. For further weight loss, drilling holes in the thinner section can be effective, but great care must be taken not to weaken the wheel to the point that it will warp and/or deform under load. If you do not have the facilities to machine a billet of material to form a wheel this way, you can create the same shape (or a drum shape) using welded fabrication methods. We recently manufactured a pair of aircraft grade aluminum wheels which started out as 5 lb solid disks, then were machined out on the lathe and then drilled on the mill. A durable solid rubber tire was then bonded to the aluminum wheel, with the finished product weighing just 1.5 lbs each — the potential weight savings are clear to see.

Rubber tire material is bonded to the aluminum wheel with carefully selected epoxy adhesive.

Assuming you are reading this and you have access to a complete engineering workshop that gives you the option of using bespoke wheels machined to your exact specifications, or fabricating hollow wheels out of any material you like, or manufacturing custom hubs and adapters to suit any commercially available wheel, what should you do? Should you make your own or buy some ready-made wheels? There really is no wrong answer to this. You may enjoy making parts so much you will invest days creating a rolling work of art when a commercial version would bolt right on, or perhaps you don't see any benefit in spending many hours to marginally improve on something that does not cost so much and could be delivered to your door overnight. As with so many things in the robot combat world, every builder will have their own preferred solutions, and if a solution works, it can always be argued that it is the right solution. SV

MANUFACTURING: Creating Composite Combat Bots

● by Bradley Hanstad of Team Think Tank (**www.twitter.com/teamthinktank**)



VDF, a 150 gram robot made by Tim Bouwens. The robot features an outer shell made of 2x2 twill weave carbon/kevlar hybrid fabric with a pure carbon fiber laminate for an upright, and a small diameter carbon tube inside for strength along the back.

The world of plastics and composites is considered a space age science, yet the industry started in the 1930s. Over the years, the composite industry has evolved and so has the amount of information. Due to such a vast and wide range of techniques, material options, and terminology, I will be condensing this data down considerably. Since this is aimed for combat robot enthusiasts, we will limit what gets explained to a smaller scale focusing on the work Team Think Tank has done in the sport. The basic elements of composites which will be covered include wet layup and vacuum bagging, leaving out infusion, prepreg usage, autoclave forming, etc. However, Team Think Tank does have their own term deemed the "auto-clave" which is essentially a car parked in the sun on a hot day. Sometimes when you simply don't have a clean room shop with an unlimited budget and all the tools in the world, you just need to improvise.

Team Think Tank (TTT) consists of myself (Bradley Hanstad), Ted Shimoda, and several others over the years that have chipped in to get the team to the point we are now. We have developed into the only active robotic team using composites to the extent that we are. Our full line of robots from 150 grams to 220 lbs have numerous composite manufacturing techniques employed and various methods of using materials from the industry in an attempt to create some of the most destructive — and the coolest looking — combat robots in the sport. As we are in the process of rebuilding our heavyweight VD6, expect to hear more from us in the future. There will be an opportunity to delve deeper into secrets of our technical abilities and skills used in building such a large robot that you can only learn

VD6 — TTT's 220 lb heavyweight robot — showing off its outer carbon fiber layer, as well as some added humor for all wedges.

over several years.

Before completely diving in, one must consider the advantages and disadvantages of using composites as a body structure, covering plates, or mounts for a combat robot. In its proper state, carbon fiber is 7-10 times stronger than steel by weight. You can make some truly lightweight yet super strong frames with composites in any shape you can possibly imagine. It is not the ultimate utopian material, however, as it can be very messy to work with and expensive to get into. Once you cut into a laminate, you greatly reduce its overall strength. With robots like Last Rites for heavyweights, it is a challenge to protect your beautiful robot. It can take some trial and error, but you can create some amazing pieces.

VD5 — TTT's 120 lb middleweight — being built for RoboGames '09 where it won 2nd place. Notice all the yellow kevlar inside.The robot has a top shell to protect the precious insides.

Metal simply can't be shaped as easily into complex forms while remaining as strong as a properly balanced laminate. Metal also transfers impact throughout the robot and tends to bend or break when hit. Composites — if used appropriately — can absorb a massive load before failure and completely bounce back to its original shape. It does have a breaking point and if you hit that, expect massive failure with delamination and a snapped laminate. Proper application is the key to success as with any material.

Now on with the basics! Composites in the form I have talked about thus far involve a fabric, a resin, and a surface to allow the system to cure on which gives it shape (always use cure when talking about resins as they do not dry or set up due to loss of water as it is a pure chemical reaction). When the resin cures, the laminate is classified as a plastic since the resin is a plastic material, yet the fabric is the strength behind it. Getting the proper amount of cloth to resin is key to obtaining a laminate even worth putting into the combat box.

VD3 — TTT's 30 pounder — is one of our first robots made from a mold to allow us to produce several of these kevlar packed bots. Expect to see a few of these VD3s fighting at RoboGames 2011 (multibots!)

In classes, we learned the magic ratio is 60% cloth to 40% resin, however, we try to work down the amount of resin to aim for more of a 70/30 ratio. The key is to make sure the entire cloth is wet out, yet not resin rich. If the system isn't wet out, it will not hold form and will be very weak. If the system is resin rich, then that section will be extra weight and brittle. Once you crack that resin rich section, the entire system has a chance of massive delamination of layers. So, how do you get that perfect ratio? It is all about weighing. Get yourself a scale now as it will be paramount in proper laminations.

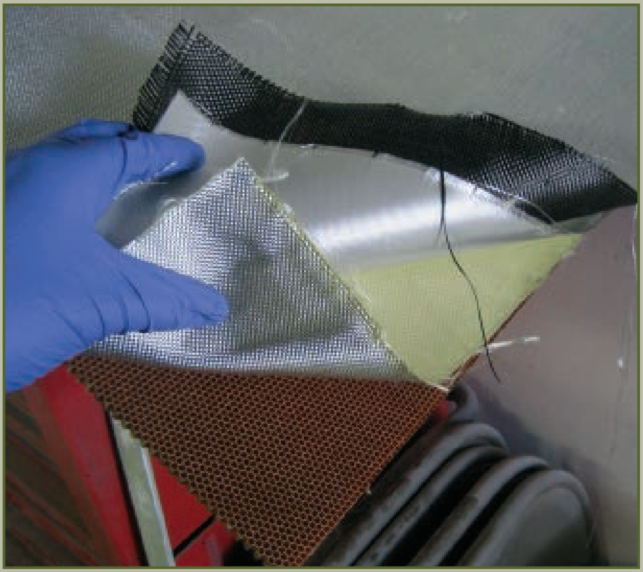
The most important component in these laminations is the fabric. So, the question is which fabric? The three main types of fabric TTT uses include fiberglass, carbon fiber, and kevlar/aramid. Beyond just the base elements, the fabrics come in a dozen forms. When you just have long strands of the material, they may be called threads; when you start bunching them up, you get continuous strand roving which may be used as chopper gun spools. Chopping up those strands and forming it into a mat of loose fibers gets you chopped strand mat; keep the strands together and weave them into very thick bi-directional patterns, you get woven roving which may be referred to as tooling cloth (or cloth made for making molds or building thickness quickly). You thin out the roving and get down to the 10 oz by square yard, and you get plain weave fabric. Cut

and stitch that into a band and you get tape. If you make the bi-directional fabric super lightweight, you get veil; take that bi-directional fabric and change the weave to a 2x2 or 4x4 pattern and you can get twill which easily forms around any shape. The forms of fiber are ever growing.

TTT's robots are generally heavily built with as much kevlar as we can get our hands on as it is very light but super absorbent to impact. It can be harder to wet out and get proper lamination but it is such a wonderful material to work with because it doesn't itch when you sand it. However, being so abrasive, it is very difficult to cut.

Carbon fiber is our next most used fabric and it is generally used for strength and stiffness in key areas. We tend to use a lot of tape forms of carbon to keep narrow segments straight. We also are guilty of using it on our final layers on the outside of the robot to give a sexy black exterior that stuns the other robots into submission.

Fiberglass is the cheapest and most readily available material, and TTT does use it heavily when we run out of kevlar or need strength and thickness. TTT also makes prototype parts with fiberglass to make sure the part will be successful before wasting kevlar, as well as using it for the molds we create for such parts. We sometimes come across remnant materials which may be specialty cloths with Mylar coatings, special weaves, or even hybrid fabrics that include carbon fiber woven with kevlar, but these materials are rare and not quite right for the average robot builder.



From bottom to top, you have a phenolic honeycomb, kevlar with a Mylar/aluminum treatment, fiberglass, carbon fiber, and finally another layer of fiberglass. All these fabrics are bi-directional. When you have your entire layup schedule together, you refer to it as a kit.

Beyond just the fabric, you then get into core materials which are designed to add thickness, insulation, structure, or form. Various types and densities of foam, honeycomb, wood, or pretty much anything that can be wet out and jammed into a laminate can be core material. Then, when you start adding resin to the mix you might have gaps or need special surfacing which may require you to add powders or materials directly to the resin before application.

Sawdust, micro-balloons (glass spheres), chopped up glass, carbon, or kevlar may be added to the resin to be used for a wide range of tasks. The beauty of composites is the flexibility of not only the standard laminate, but the array of materials in which you can use them.

There may be a lot of information about various fabric materials that can be used successfully in combat robots, but as far as resins go there is one clear winner: epoxy. Polyester, vinyl ester, phenolic, polyurethane, and bismaleimide resins just aren't quite capable of being what epoxy is. Epoxy may be a bit more expensive, but you just can't beat its strength and ease of mixing without using dangerous catalysts.

Most epoxy types have a standard two part system, but make sure you read the exact ratio of hardener to epoxy as per the manufacturers instructions (as well as the MSDS). We have used various epoxies at various price levels and have found a lot of epoxy manufacturers put plastesizers and sometimes just a whole lot of junk that adds nothing to the strength of the resin but merely increases the volume.

Max Bond — which is referred to as a marine grade resin — is one of the best quality resin systems

Max Bond epoxy system from Polymer Composites, Inc. This would be the marine grade resin two gallon kit as purchased through eBay or directly through the company. It is a 1:1 ratio mix of part A and part B.

you can get. Through eBay, it can be found from the seller “polymerproducts.” This resin does have a slight tan coloring to it, but if you need a crystal clear resin they also supply a resin labeled Max CLR which is a very strong resin made for visual clarity. Both of these systems are a 1:1 ratio of part A and part B by weight.

Now that you have the meat of composites, we’ll need to go over the preparation and cooking. Next month, I will walk you through a very basic lamination by making a flat panel, and will go over all the tools, techniques, and safety precautions you should use. SV

EVENTS

Completed and Upcoming Events

Completed Events: June 1 - July 5

The Edge was presented by QRSC in Brisbane, Queensland on June 26th. Fifteen bots were registered.

Gulf Coast Robot Sports 5 was presented by Gulf Coast Robot Sports in Bradenton, FL on June 5th. Nineteen bots were registered.

Upcoming Events: September - October

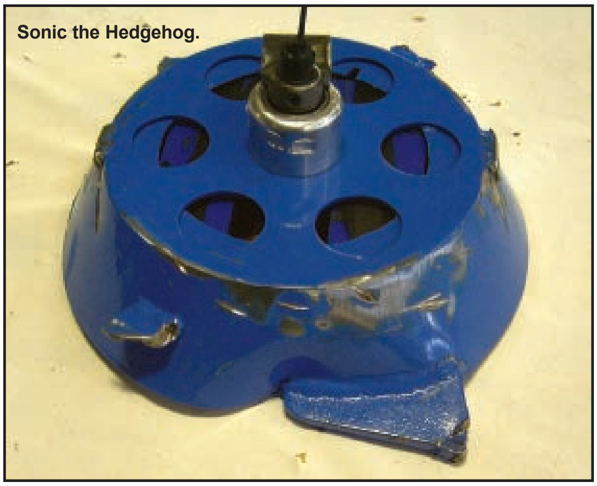
HORD Fall 2010 will be presented by the Ohio Robot Club in Akron, OH, on September 25th. Go to **www.ohiorobotclub.com** for more information.

Robothon Robot Combat 2010 will be presented by Western Allied Robotics in Seattle, WA on October 24th. Go to **www.westernalliedrobotics.com** for more information. SV

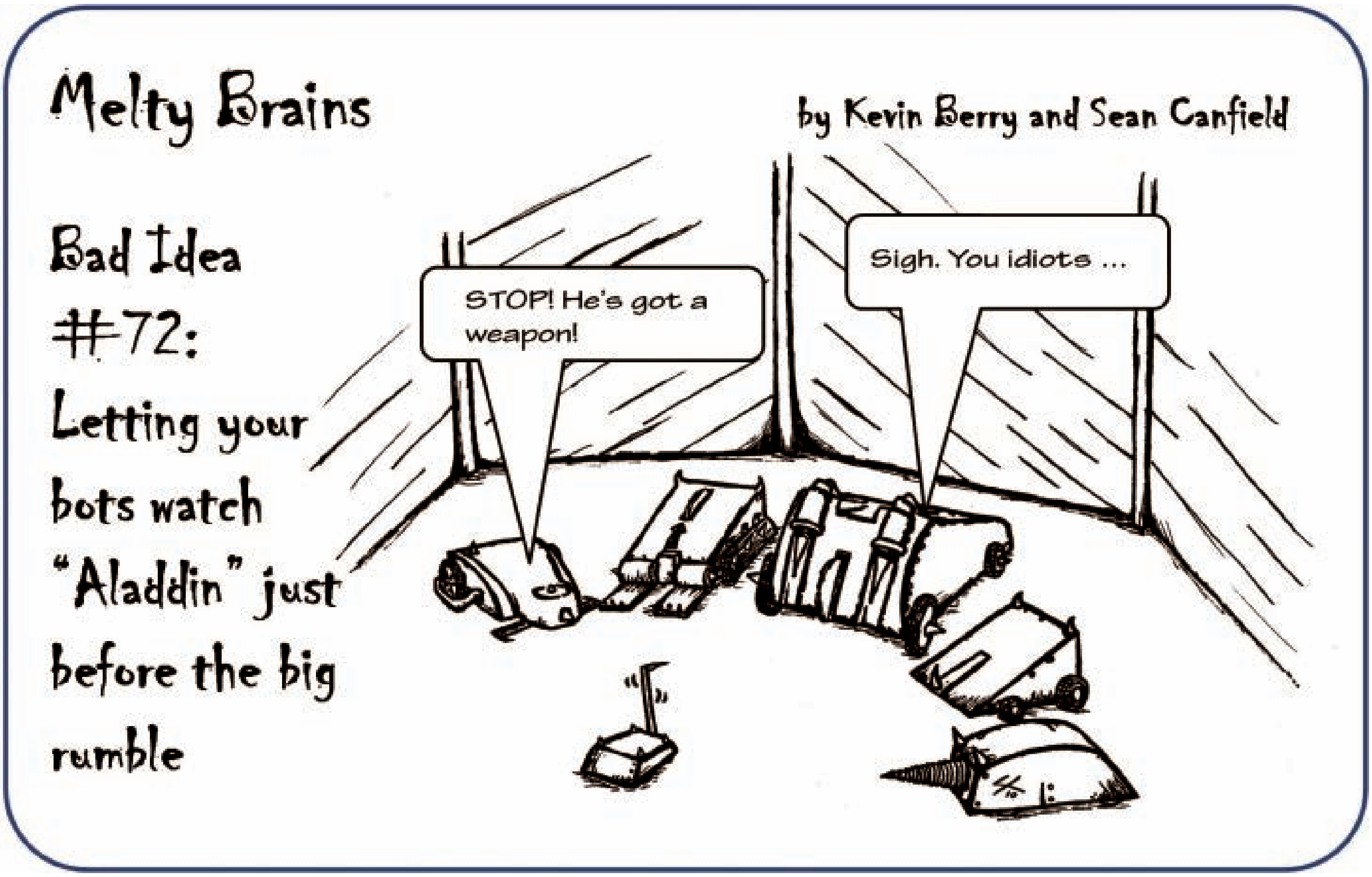
COMBAT ZONE’S GREATEST HITS

● by Kevin Berry

Dalton O’Conner of Team Massacre Robotics sent in this one. The bot is Sonic the Hedgehog from the 12 lb class. The weapon is 14” diameter, 1/8” rolled steel shell with 3/8” teeth, spinning at


Sonic the Hedgehog.


Sad Sonic.



about 5,000 RPM. His tale:

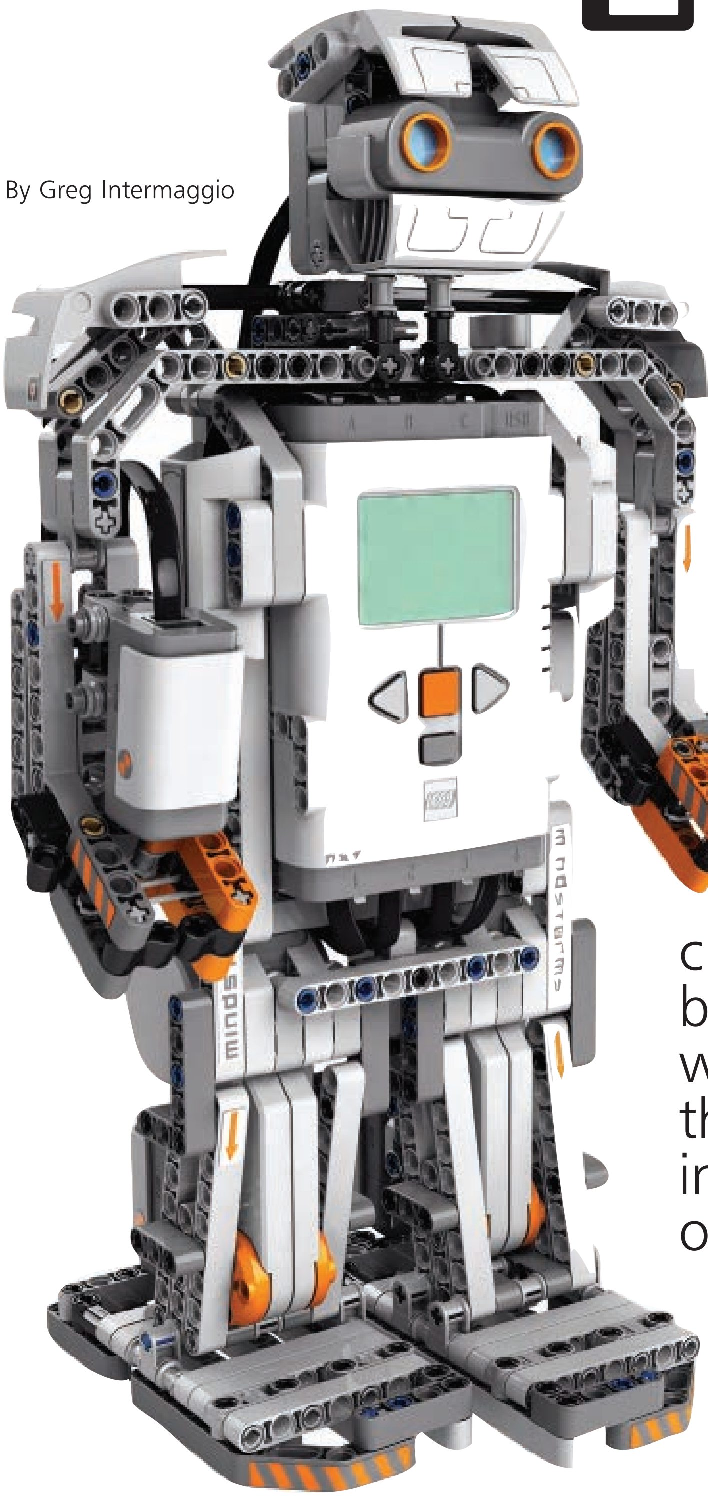
"The fight was at Motorama 2010 vs. Surgical Strike. Pete (Smith) actually reversed the weapon thinking Sonic spun the other way, hoping we would glance off each other instead of what actually happened. Surgical Strike came out of the arena hurting (at least). Our mild steel tooth is fractured about 5/8" back, and the welds on both teeth are tearing off the surrounding material. Everything still worked after though!"

See the video "Sonic the Hedgehog vs. Surgical Strike" at **www.youtube.com/watch?v=2WU5rLNJ4xo**. **SV**

Before and after photos, brief description of the fight, and builder's name can be submitted to me at LegendaryRobotics@gmail.com. Or, if you have an action shot that clearly shows what's going on, those are welcome too! These don't have to be current, anything you can (legally) submit — clear back to the Good Old Days of wooden bots and iron builders — is fair game.

The NXT Big Thing #2

By Greg Intermaggio



Getting Touchy-Feely

In this ongoing series of articles, we'll take a closer look at the LEGO MINDSTORMS NXT, and all the features that make it great.

Last month, we started by creating a simple driving chassis and doing some basic programming. Now, we'll move on and learn the computer software included in the kit, and use our first sensor to make a robot that can navigate around a room.

Let's begin honing our skills by answering the following questions:

- Why use the NXT computer software?
- How can we get the most out of our touch sensors?
- What other sensors can we use in the future to continue the development of Eddie?

After we cover these questions, we'll dive in to some programming and extensive use of the touch sensor.

Why use the NXT computer software?

While the NXT itself has a very basic programming function built in, it is almost nothing when compared to the power and versatility of the computer software. While the built-in function allows for a simple, linear, five-step program, the computer software allows for non-linear programming with no limit on the number of steps. In short, if you're not using the computer software, you're not getting the most out of your NXT.

How can we get the most out of our touch sensors?

Touch sensors are nifty for several reasons. First and foremost, they present a way for our robot to actually 'feel.' In other words, touch sensors give our robot a way to interact with its environment in a directly physical way. As opposed to something like an ultrasonic sensor that would let us see how far away an object is, a touch sensor allows us to push into things, making direct contact.

In addition to the physical interaction aspect of the touch sensor, it has a distinct advantage of simplicity over other sensors. The touch sensor's output value is binary — meaning that it will always be either 1 or 0. This binary output is very easy to work with, especially when you consider the implications when working with multiple sensors. Basically, you can add together the output values of multiple sensors, and at any given point know exactly how many sensors are being tripped.

The ability to physically interact with the surrounding environment — coupled with the simplicity of the sensor's output value — make the touch sensor both a great asset in developing robots in general, as well as a great stepping stone into more complex sensors.

The MINDSTORMS NXT kit has several variations, and within them, several software variations. This article is based on the NXT kit available from LEGO Education, rather than the retail version. Additionally, the software used in this article is the MINDSTORMS Education NXT Programming version 2.0. That said, if you have a different kit or different software, have no fear — chances are you'll be fine. Just know that you're on your own with technical issues.

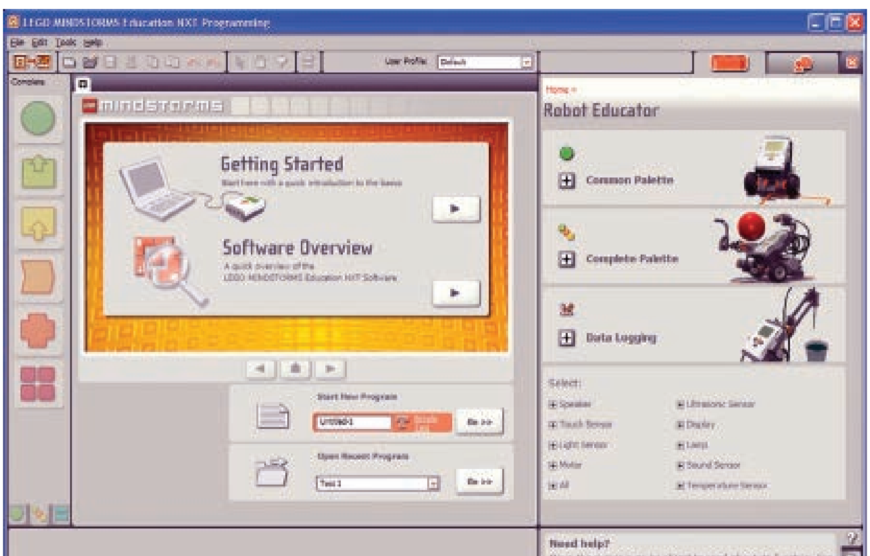
What other sensors can we use in the future to continue the development of Eddie?

The awesome thing about working with the NXT is that there is truly no limit to what sensors you can add. There are a few sensors available from LEGO directly, as well as a wider selection from HiTechnic, and an even wider selection of precision sensors from Vernier. If you're feeling truly adventurous, you can even buy a kit to develop your own sensor and accompanying programming block within the NXT's software.

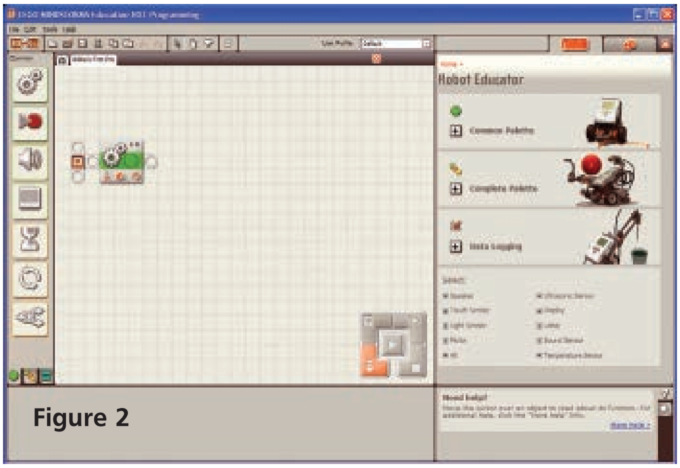
With that in mind, let's get started by familiarizing ourselves with the NXT computer software.

Software Basics

After you've installed the software on your computer, opening it will bring you to a welcome menu including a window in the center with both "Getting Started" and "Software Overview" options. If this is your first time using the software, I highly recommend going through each of these guides, as well as familiarizing yourself further with the NXT software by running through a few of the "Robot



The Software



Educator" exercises available on the right side of the screen.

With the basics down, let's move on to programming Eddie from the computer for the first time.

Eddie's First [Real] Program:

- Open the NXT software on your computer.
- Name your program "Eddie's First Program" in the text field below "Start New Program" before clicking "Go >>".
- Our first program will be just one step long with a programming "block" from the "Common Palette."
- From the Common Palette, click and drag a "Move" block as the first step of the program.
- Click on your newly placed move block and make sure its attributes match those of **Figure 1**.
- If your program looks like **Figure 2**, turn on your NXT, connect it via USB, and click the "Download" button indicated in **Figure 2**.
- Next, disconnect Eddie from your computer and navigate to My Files > Software Files > Eddie's First Program.rbt.
- Put Eddie on the floor and press run. If all goes well, he should briefly roll forwards before stopping and terminating the program.

Quick Recap

We just did a quick first program in the NXT computer software both to introduce the software and make sure that our robot is configured correctly. Remember: Because our gear train is reversing the direction of the motor output, making a motor go forward means the robot will go backwards, and vice versa. That's why by programming the motors to go backwards, the robot ultimately moves forward.

Now that you understand the basics, let's get right into things by adding a touch sensor. First, you'll need to build the touch sensor attachment. Follow the instructions to get it ready for action.

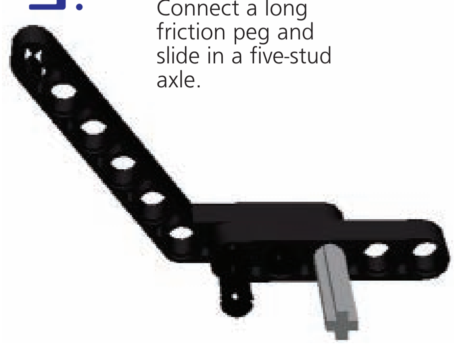
Touch Sensor Attachment Building Instructions

NOTE: This attachment snaps easily in and out of the front of Eddie — the robot we designed last month. You'll have a choice between a lower height so Eddie detects smaller objects, or a higher height — encouraging Eddie to drive right over small obstacles.

1. Start with a 4 x 6 angular beam and a friction peg.

2. Add a five-hole studless beam.

3. Connect a long friction peg and slide in a five-stud axle.



4.

Attach the touch sensor to the assembly.

5.

Take a long perpendicular axle joiner.

6.

Snap in two long friction pegs.

7.

Attach the axle joiner assembly to the main assembly.

8.

Slide in a five-hole studless beam.

9.

Attach a three-stud axle to the back, and a standard friction peg to the side as indicated.

10.

Push two standard size perpendicular axle joiners onto the rear axle.

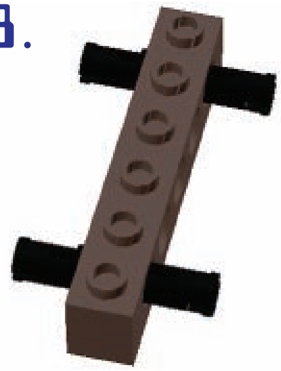
11.

Slide a five-stud axle into the axle hole above the touch sensor, and snap in two long friction pegs with axle holes.

12.

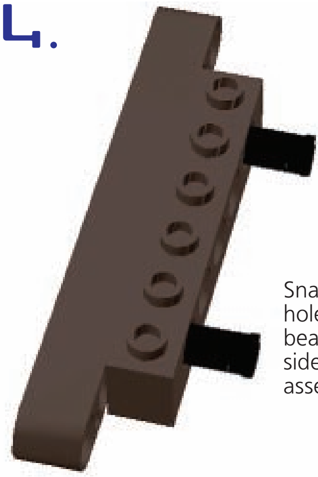
Find a six-stud beam.

13.



Attach two long friction pegs.

14.



Snap in a nine-hole studless beam on one side of the assembly.

15.

Put another nine-hole studless beam on the other side.

16.

Slide this assembly onto the five-stud axle on the main assembly.

17.

Attach a 4 x 6 angular beam to close it off and complete the touch sensor attachment.

Stopping on Contact

With the touch sensor attachment complete, it's time to write a program that utilizes it. Make sure to affix the attachment to the robot as indicated, and plug it in to sensor port 1 before going on.

- Create a new program called "Touch 1."
- Drag a Move block in as the first step, and configure it to move motors B and C backwards for an unlimited duration (**Figure 3**).

Figure 3

Figure 5

Figure 6

- In the “Wait” menu, find “Touch” and drag it into step 2 (**Figure 4**).
- Make sure it’s configured to wait to be pressed, and set to port 1 (**Figure 5**).
- Add a Move block that stops motors B and C (**Figure 6**).
- Save your program, then download it to the NXT and test it out.

Eddie should drive forward until he hits a wall, tripping the touch sensor and stopping the motors to end the program. With that program under our belt, let’s take this even further.

Avoiding Obstacles

It’s time to learn how to make Eddie avoid obstacles. Start by opening your “Touch 1” program, and saving it as “Touch 2” — that will preserve the Touch 1 program while making a duplicate that we can modify.

- In the Touch 2 program, select the last step, where we stopped motors B and C. Rather than stopping when he hits a wall, we want Eddie to back up and turn away. To do that, we’ll change this step to spin the motors forward for two seconds, thus making the robot move backwards. Confused yet? Check **Figure 7** to see what the code should look like at this point.
- Now that Eddie will back up, we need to make him turn briefly so he avoids the obstacle. Add another Move block, and set motors B and C to go forward with the steering bar

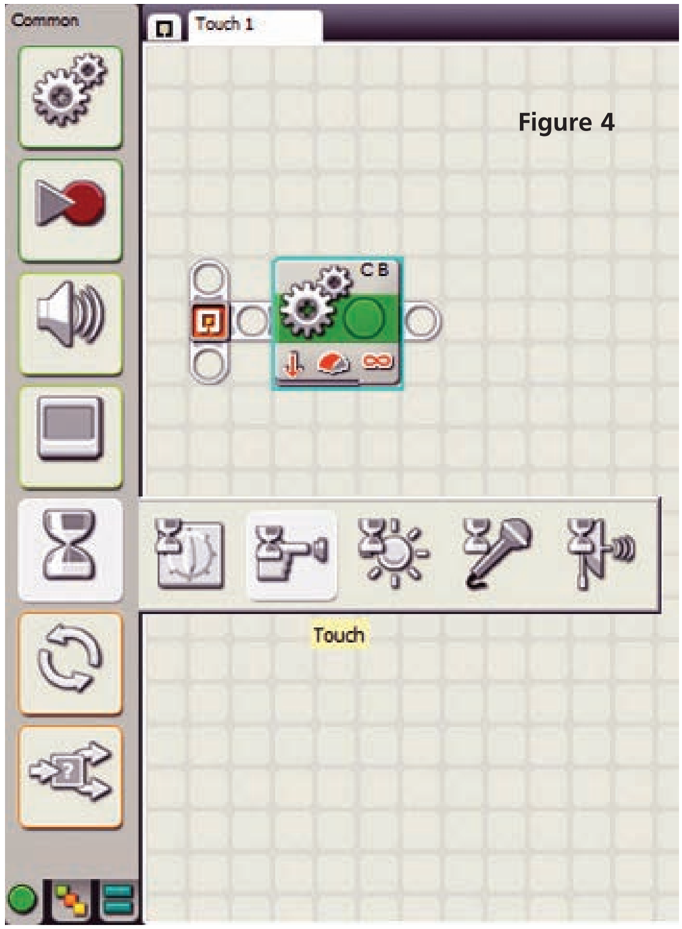


Figure 4

Figure 8

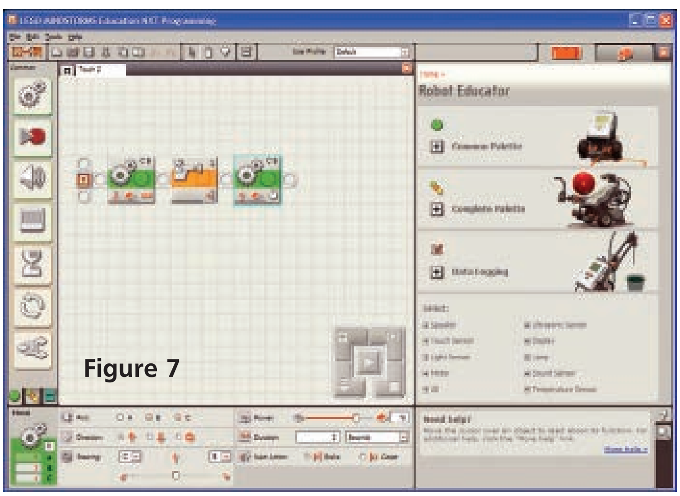


Figure 7

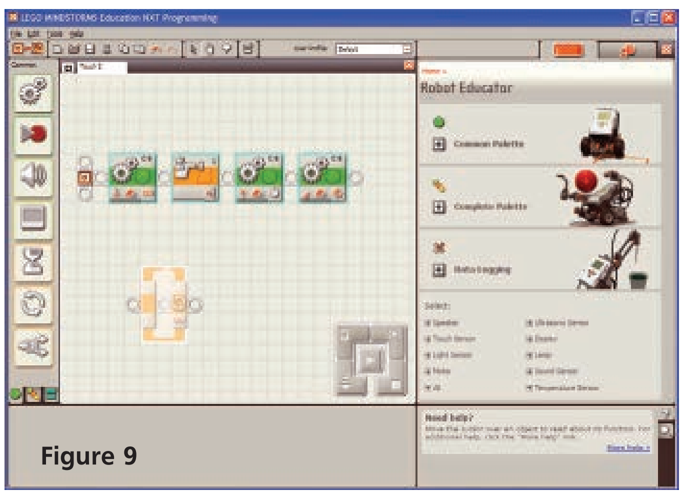


Figure 9

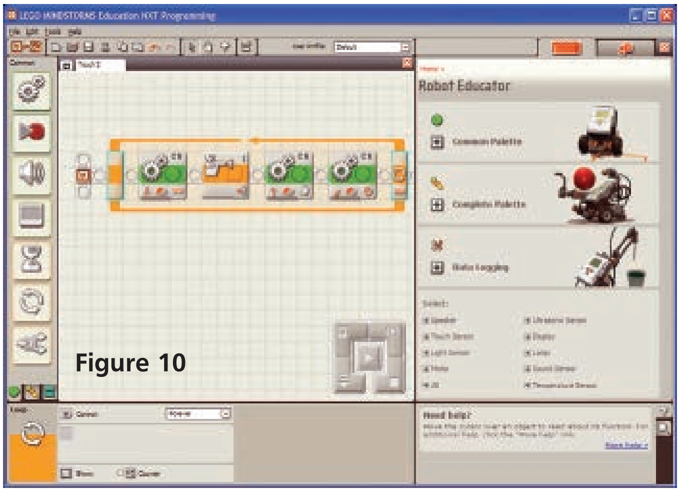


Figure 10

all the way to the left for one rotation (**Figure 8**).

- For the last step, we need to add a loop which will cause the program to repeat itself indefinitely, rather than stopping once it performs each action once. Drag a loop into the programming area but off to the side of the rest of the blocks.
- Click and drag across the four programming blocks aside from the loop to select them all at once (**Figure 9**).
- Drag the four selected programming blocks into the loop, and connect the loop to the beginning of the program (**Figure 10**).
- Save the program, and give it a test run.

Eddie will move forward until he hits a wall that trips his touch sensor, causing him to move backwards, turn away from the obstacle, and then will continue onward. You've just built and programmed a robot that is capable of interacting with its environment physically via touch sensor, and reacting to it by avoiding obstacles!

Wrapping Up

Next time, we'll be looking at two more sensors, and getting into some more complex programming to really get a feel for just how powerful the NXT is.

If you're enjoying The NXT Big Thing, and want to see more articles featuring the MINDSTORMS NXT, please take a second to let *SERVO* know at editor@servomagazine.com.

Thanks for reading — see you next time! **SV**

Tool Time With the Versapak Smart Charger

By Richard Spelling

While researching a project (yes, I have more projects than I have time ... who doesn't?), I decided I needed to learn a little bit about charging NiCd and NiMH batteries. Now, that was some very interesting reading. In particular, it was mentioned by several sources and manufacturers that NiCd and NiMH batteries could be charged 500 to 1,000 times.

Now, wait a second here. Back in the day, I was enamored by the thought of interchangeable battery packs, and I bought quite a few Versapak tools. The batteries that come with them say they are good for up to 300 charges — not 500 to 1,000. My own experience, plus anecdotes from friends that use them, would lead one to think that they are good for MUCH less than that. In fact, I have had the experience of buying new Versapak batteries, using them once, sticking them on the charger, then when I went to use the tool again, the battery would be completely dead.

I slowly came to the realization that the chargers for the Versapaks burn up the batteries. If you leave them on the charger — which is a natural thing to do in order to keep them "topped off" — you will ruin the batteries.

The current that battery chargers charge with is generally expressed as a fraction of the capacity of the battery, denoted as "C." So, for instance, the standard NiCd

Versapak battery packs are rated at a C of 1.2 amp hours. Therefore, a charge rate of C/10 would be a charge current of 120 mA. Some people claim that continuously charging NiCd batteries at C/10 (which is what the stock chargers do) won't hurt them; others claim that it will shorten their lives, but does not cause other problems.

I would say from both personal experience and talking to friends, that it will DRAMATICALLY reduce the life of the battery packs. So much so that you should never *ever* leave these batteries on the chargers after they are fully charged (which is probably in the instructions I didn't read ...).

Of course, NiCd batteries also "self discharge" by up to 10% per month which means when you go to use the batteries that you have taken off the charger to keep them from being killed, there is a very good chance they will need charging. You can't win! Or can you?

I decided that I was going to build a smart charger for the Versapak batteries. These are chargers that detect when NiCD or NiMH batteries are fully charged, and shut down the charging current. While some so-called smart chargers charge two cells in parallel, I didn't think this was such a good idea. So, I decided to build a smart charger for each individual Versapak battery pack, and do it all inside the standard holder/charger. Yes, I would be voiding any warranty on the charger and probably the batteries, but it would be fun (and useful) to build, so I didn't care. BTW, these battery packs are really just three sub-C 4/5 size batteries, but that could be a whole other article!

I wanted to be able to use the existing holders for the Versapaks, and after taking one apart I discovered that there is plenty of room inside for additional circuit boards (or even expanding the component count and size of the current board). The battery contacts need a board to hold them in place, so I would need to have one under the batteries where the original board was anyway. I decided to

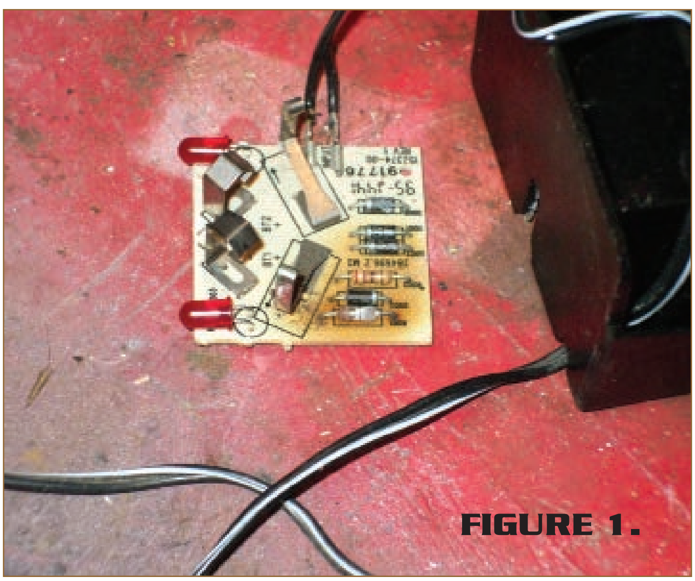


FIGURE 1.

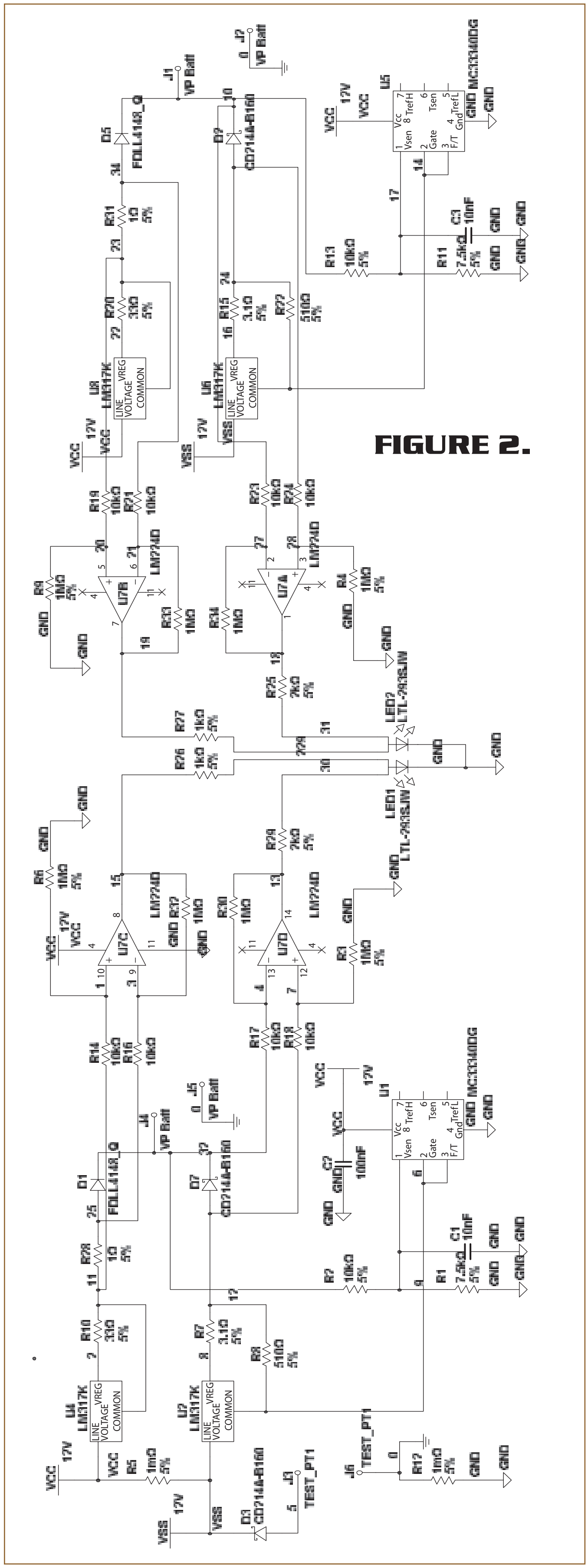


FIGURE 2.

Toner Transfer

Toner transfer is a method (common in hobby PCB making) of printing your circuit board on paper, then using heat and pressure to de-laminate the plastic toner onto the clean, bare copper untreated (and cheap) PCB.

It can be done with ink jet photo paper and a clothes iron, but you get more consistent results if you use a laminator and glossy paper. I use Hammermill Laser Gloss, but others have had good results with the gloss paper used in magazine pages.

You print the resist pattern on the paper, use the iron or laminator to transfer it to the copper, then you soak the paper in water to make it easy to remove. The plastic toner stays on the copper, and makes an excellent etch resist.

It takes a little practice to get the technique down, but once you have it you can knock out good prototype boards for essentially the cost of the copper board.

see if I could build a better charger (with two smart charger circuits) that would fit in exactly the same slot as the current printed circuit board (PCB); see **Figure 1**.

Cramming all these components in this tiny space meant I would have to use surface-mount components. That's right up my alley anyway, as I use them exclusively for all the circuit boards I make. This has the drawback that I can't prototype things as easily as with through-hole components and a breadboard, but there is the advantage that when I have a working product, I don't have to convert it to the final form. I just have to stick it in a case and move on. I do my prototyping using software design and simulation tools, specifically NI Multisim (a descendant of Electronic Workbench), and minimize my breadboard costs by buying in bulk and making my own PCBs. SMD components are much less expensive than through-hole parts.

There are basically two methods of smart-charging NiCd and NiMH batteries: one method detects the fall in voltage that happens when the battery is fully charged and is called delta V; the other method detects the rise in temperature of the fully charged battery and is called delta T. I decided to forgo the temperature change method of determining that the batteries are charged, and instead used the delta V method. This way, I didn't have to fiddle with trying to sense the temperature of the batteries in the stock Versapak battery holder. Sensing the temperature of the batteries would require putting thermistors in physical contact with the batteries, and I didn't want to have to make the modifications to the holder that this would entail.

There are quite a few manufacturers of smart charger chips, but Mouser (my favorite electronics supplier) carries the MC33340DG made by ON Semiconductor which happens to use the delta V method for primary charge termination. It also supports elapsed time or even thermistor temperature sensing as secondary methods for terminating charging. And at $1.45, each they are fairly inexpensive.

The manufacturer's example circuit shows a trickle mode, as well as a fast charge mode. Cool! So, the charger I was building would be a fast charger, then when the batteries were done charging, I could leave the batteries on the blasted thing, and it would keep them fully topped off.

As you can see from **Figure 1**, the existing board has hardly any components on it at all. I decided to use bi-color LEDs, so that I could get an indication of when the charger switched from fast to trickle mode, and so I could use the existing plastic case with virtually no modifications.

As these things tend to work out, the charger section worked the first time, but the circuit to light the LEDs didn't work at all; the trickle LEDs always stayed on. I had come up with the bright idea of detecting the forward voltage drop over the blocking diodes in order to light the corresponding trickle and fast charge LEDs. Testing showed this only sort-of works. I could do it with the fast charge LED, but the trickle LED showed 0.5V forward voltage across it just from various leakage currents, and, of course, the standard 0.7V(ish) forward drop when the circuit was trickle charging.

Also, I had originally thought to use comparators to trigger the LEDs. This worked fine for the fast charge LED, as the inverting input would go higher than the non-inverting input when it switched from fast to trickle. However, the inverting input for the trickling LED would never be higher than the non-inverting input, so the trickle LED would always stay on.

After thinking about this for a bit, I realized I could simply use differential amplifiers with a specific finite gain (see **Figure 2**). As I was getting 0.5V and 0.8V across the trickle circuit diode, I decided to stick a current sensing resistor on the trickle side and measure that instead of the

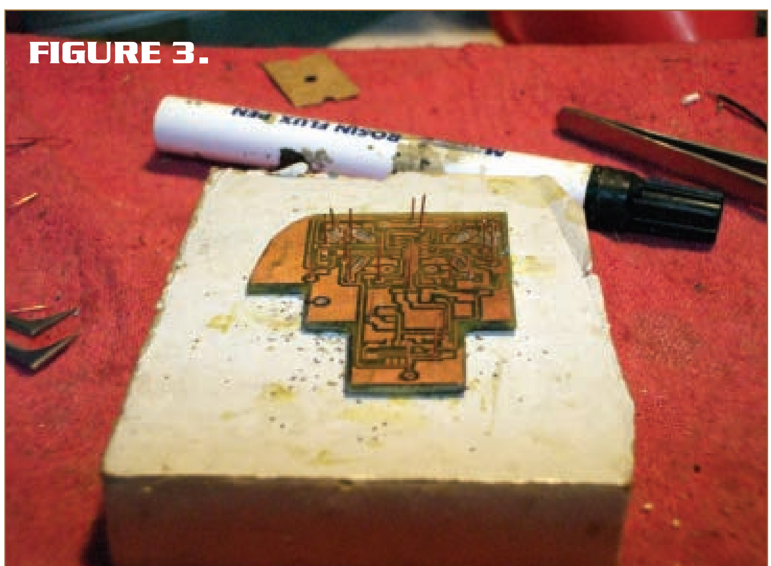


FIGURE 3.

forward drop over the diode. This gave me 35 mV when trickling and 00.6 mV when no battery was connected. I turned the gain down on the differential amps so that the amplified voltage in the off state would be less than the forward voltage drop on the green section of the bi-color LED.

Don't have Versapaks? The charger circuit in **Figure 2** can be used to charge any NiCd or NiMH 3.6V (three cell) battery pack without modifications, or any two cell pack (which I'm not sure anyone makes). It won't work beyond that on either end without changing the voltage sense divider. The voltage divider for the smart charger chips has to have the sense voltage in a certain range (1.0V to 2.0V for the chips I use). It is also more sensitive and works better if you use the high end of the range (mine is set for ~1.85V). Commercial "universal" NiCd/NiMH chargers you can purchase work from 7.2-12V, so they work on the same principle.

If you modify the circuit in **Figure 2** so that the voltage divider formed by R1/R2 (and R11/R13) gives you around 1.85V when your battery packs are fully charged, you can use the circuit to charge any NiCd or NiMH battery pack up to 18V. This is the most the MC33340DG will handle. Of course, you could isolate the MC33340DG and the LM224 behind voltage regulators and use high voltage adjustable regulators (like the Texas Instruments TL738) then go quite high with the voltage, but that is beyond the scope of this article. I used the toner transfer method to make the board shown in **Figure 3**. I can reliably make boards with 7 mil traces and 10 mil spacing, though, truth be told, the toner does spread out a bit in the process, so the traces are slightly wider and the gaps narrower. I'm more comfortable using a 15 mil trace clearance, but in order to get this thing to route on the tiny board I had to bump the clearance down a bit.

I have had a few of these boards professionally manufactured, if any of you want to build your own Versapak smart chargers without making your own board. They would be available for a few dollars (cost would depend on order size). If you are interested, just send me an email at nutsandvolts@chebacco.com to ask about them. A few words about working with surface-mount components:

- A 1206 form factor is about as small as you want to go; this is about the size of a match head. With the proper tools and good magnification, you can work with considerably smaller SMT components. I have worked with 0603, which is about the size of two grains of salt. Others work with even smaller parts. However, the 1206 form factor is large enough to have a decent power rating, heavy enough to not stick to the tweezers, and almost solder-able without magnification. Plus, it's about the same price as the smaller parts.

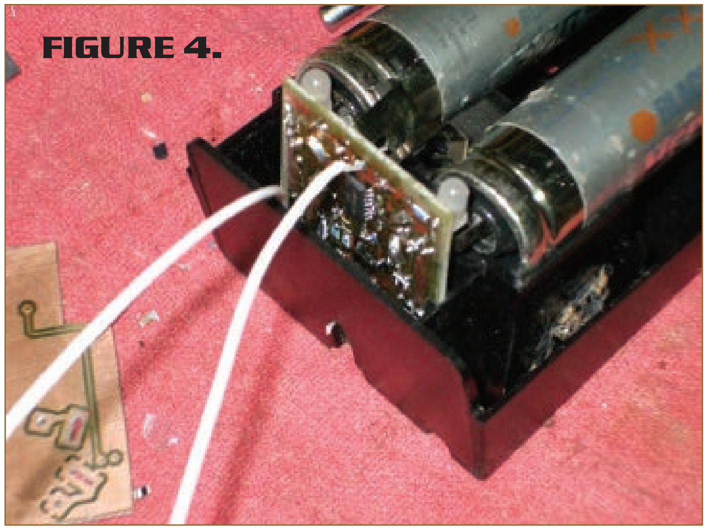


FIGURE 4.

- Use lots of flux. I like to coat the bare board with flux, let it dry, then give it another coat just before I solder on the components. Surface tension is your friend. I use a rosin core flux pin exclusively.
- Use magnification and lots of light.
- A temperature-controlled soldering iron is best (set to 350°C), but you can get away with a small 15W iron with a fine tip.

Parts List

Sacrificial (but to be greatly improved) stock Versapak charger power supply: 12VDC @1A.

Components:

QTY	DESCRIPTION
2	MC33340DG (smart charger chips in SOIC-8 footprint)
1	LM224D (quad op-amp in SOIC-16 footprint)
2	LM317LIPKG3 (any LM317 clone in the SOT-89 footprint should work)
2	LM317MDTX (any LM317 in DPAK footprint should work)
2	TLUV5300 bi-color LED

Resistors, 2010 surface-mount

2	3.1 ohm

Resistors, 1206 surface-mount, 5% okay:

8	1M
12	10K
2	7.5K
2	2K
2	1K
2	510 ohms
2	33 ohms
2	1 ohms
2	0 ohms

Misc.:

Caps, 1206 surface-mount

2	10 µF
1	100 µF

Diodes:

2	CD214A-B160
2	FDLL4148_Q

- Put everything on one face of the board before you try to solder the parts. Just get them on top of the pads and oriented close to correctly; you can fine-tune as you solder. The wet flux will hold things in place until you are ready to start soldering.
- I hold the tweezers in my left hand in such a way that the little finger of my hand is resting on the work surface. I then move the components around by moving my arm, not my fingers. (This is how artists who work on master dies for coins do it.)
- The method for SMT soldering I've found recommended on several websites — "Put some solder on the pad, hold the component in place, then melt the solder" — works okay, but I prefer the following. I just put a tiny dab of solder on the end of my iron, hold the component in place with the tweezers, and touch the iron to the joint between the component and the pad. Surface tension does the rest. Use a very fine lead free solder; .020" size works for me.
- Don't worry about having a little extra solder on the component, and don't worry about the part being perfectly lined up. Hold the iron on the part for about a half second, then pull it off and give it a second to cool before letting go with the tweezers. Solder one pad on all components, then go back and finish that side of the board. Check your work under high magnification (at least 10X), flip the board over, and repeat.
- You will need to pull the battery connectors off the old board that comes with your sacrificial charger. I've found that the best way to do this is to simply load up some solder wick with flux and get all the solder to wick-out. Then, gently bend up the tabs that hold the clip in place and then heat the leg again while gently pulling on the part.
- You will need a new power supply to run your fast charger; the one that comes with the Versapak chargers doesn't put out enough voltage (or current). Anything that puts out approximately 12V at 1A should work. I found a deal on some scratch and dent linear supplies, but a switching supply would work okay as well. Most of the 12V wall wart supplies I have tested put out more than 12V, but you should be okay with a few extra volts. I wouldn't go much higher than around 16V.
- Once everything has been assembled and you place the new and improved board into the Versapak charger/holder (**Figure 4**), you can plug in the power supply. The LEDs should flicker briefly then go out. This is an unintentional feature, but it lets you know you have hooked up the power and everything is working. So, I didn't try to fix it. When you plug a battery in, you will get an orange light (red for fast charge plus green for trickle charge equals orange) showing that the charger is working and attempting to charge the batteries. When the batteries are almost fully charged, you will see the light switch from orange to green and then back to orange periodically. This is the smart charger chip shutting off the fast charge circuit in order to get an accurate reading on the battery voltage. Given enough time, the LEDs will stay green, indicating that the batteries have been charged and the charger has changed to trickle charge only mode.

Enjoy! **SV**

The Wasp Embedded Processor Takes The Sting Out of C

By Fred Eady

Mastering the C programming language has its advantages. The power of the C programming language can be brought to bear on personal computers, microprocessors, and microcontrollers. If you're already a C expert, you'll be interested in the hardware we're about to discuss. On the other hand, if you're a hardware guru and want to know more about how C and an AVR microcontroller work together, we've got something for you, too.

A Silicon Wasp

The SOC Robotics Wasp Embedded Processor you see in **Photo 1** can't fly on its own. However, it is compact enough to be part of just about anything mechanical that takes to the air. The Wasp printed circuit board (PCB) comes in at 1.472 x 1.152 inches. Loaded with its native electronics only, the Wasp weighs in at six grams.

The Wasp's intelligence is provided by an Atmel ATmega644 microcontroller. The ATmega644 comes from the showroom floor with 64 KB of in-system self-programmable Flash program memory, 2 KB of EEPROM, and 4 KB of internal SRAM. I could go on and on extolling the virtues of the ATmega644. However, the idea behind the Wasp's design is to shield the C programmer from the ATmega644's internal intricacies while allowing the C programmer full access to the microcontroller's resources. The Wasp can also be outfitted with members of the ATmega16 and ATmega32 families.

The Wasp is as power-stingy as it is small. When it's "flying," the Wasp needs only 12 mA or so to stay in the air. When it's sitting on the nest, it consumes 0.6 mA. Like its insect counterpart, the Wasp's power consumption is dependent on how fast it is being clocked and what on-chip peripherals are powered up. This extremely low power

consumption figure allows the Wasp electronics to run powered only by a single 3.3 volt coin cell.

The Wasp design is similar in thought to that of the military Humvee. There are many roles this robotics processor can take on depending upon how it is equipped and configured. For instance, the Wasp is perfect for data logging missions as it can be programmed to store analog-to-digital readings into its internal EEPROM or external serial Flash. It can be loaded with serial Flash parts that range from 512K to 8M of eight-bit storage. The ATmega644's ability to support an external 32.768 kHz clock crystal further adds to the Wasp's data logging prowess as data logging can be tied to time of day or timed events via the ATmega644's real time clock. In addition, the Wasp can sleep until the real time clock awakens the ATmega644 to take a reading.

SPI and I²C portals are common to most microcontrollers and the ATmega644 is no exception. The Wasp uses the services of the ATmega644's SPI portal to communicate with the serial Flash. Atmel's acronym for I²C is TWI (Two Wire Interface). The Wasp utilizes TWI architecture to communicate with other I²C-based smart peripherals. The Wasp's I²C portal consists of a male Molex connector that also is pinned to power or be powered by external I²C devices. The Wasp can run at 3.3 volts or 5.0 volts. When external serial Flash is installed, it is recommended to power the Wasp at 3.3 volts to avoid releasing the serial Flash IC's magic smoke. The only way to fly the Wasp with a 20 MHz crystal is to power the silicon polistes exclamans with 5.0 volts.

Another peek at **Photo 1** reveals the presence of 8 Mbytes of serial Flash. I flipped the Wasp over in **Photo 2** to show you the Freescale Semiconductor MMA7260 three-axis accelerometer IC. The smaller IC just above the MMA7260 is a Burr-Brown DAC8501 digital-to-analog converter IC. Both the MMA7260 and DAC8501 are optional components. In that the serial Flash and accelerometer are installed, the crystal you see in **Photo 1** must be a 10 MHz part.

Photos 1 and **2** tell us what stuff the Wasp is made of. However, **Figure 1** provides a detailed view of the Wasp's I/O subsystem and power assignments. When the Wasp isn't sitting on its nest, the SPI portal on digital port J2 can be used to program the Wasp's AVR. Note that the AVR's UART interface (RXD and TXD) is located on digital port J3 along with the pinned out I²C portal (SCL and SDA).

In the old days of the EDTP ATA hard drive controller, the AVR wanted to see a 10-pin ICSP termination. These days, the latest AVRISP mkII is fitted with a six-pin ICSP termination. The Wasp I received

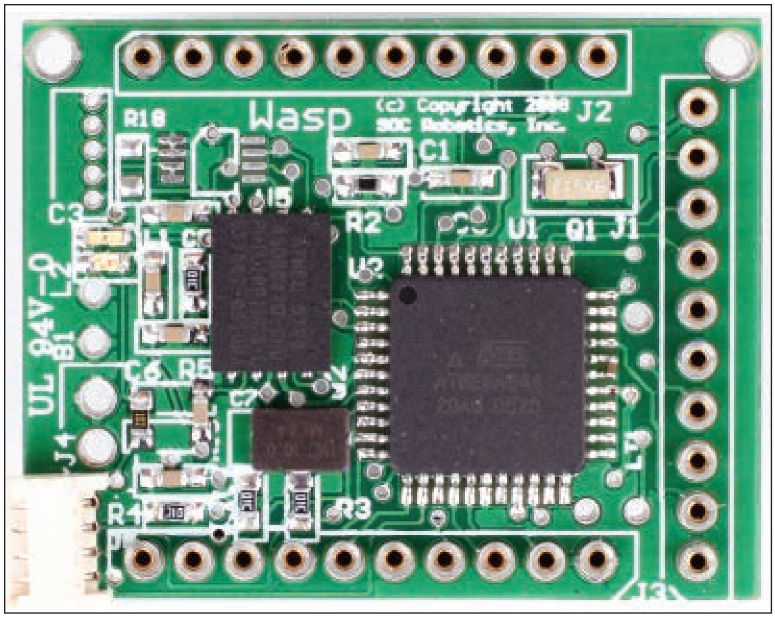


PHOTO 1. This is a Wasp. Instead of a stinger, wings, big eyes, and six legs, you're looking down on an Atmel ATmega644 microcontroller, 8 MB of Atmel serial Flash, a 10 MHz CPU crystal, a 32.768 kHz real time clock crystal, and various capacitors and resistors. The Molex connector is a powered I²C portal.

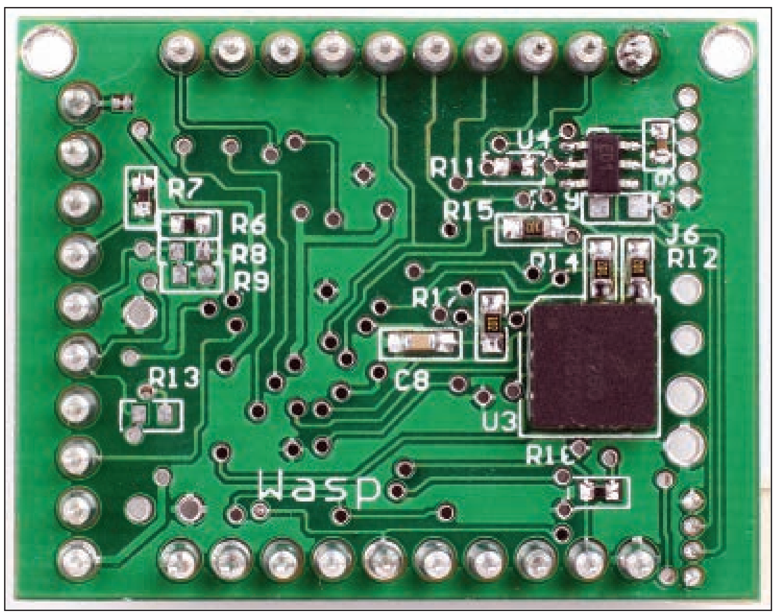


PHOTO 2. This is an underside shot of the Wasp. The Burr-Brown DAC8501 shares the spotlight in this photo with the optional Freescale MMA7260 three-axis accelerometer.

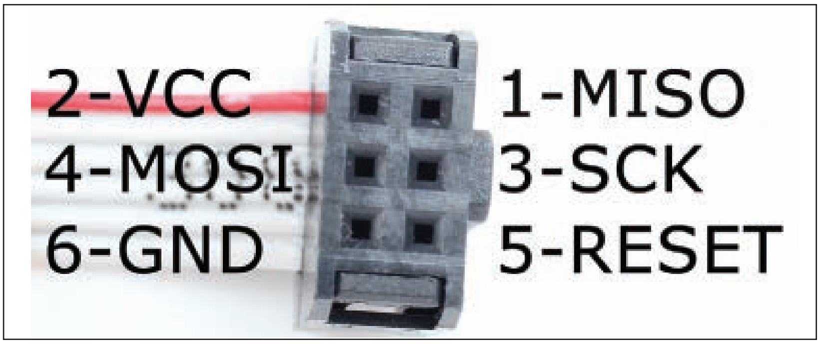
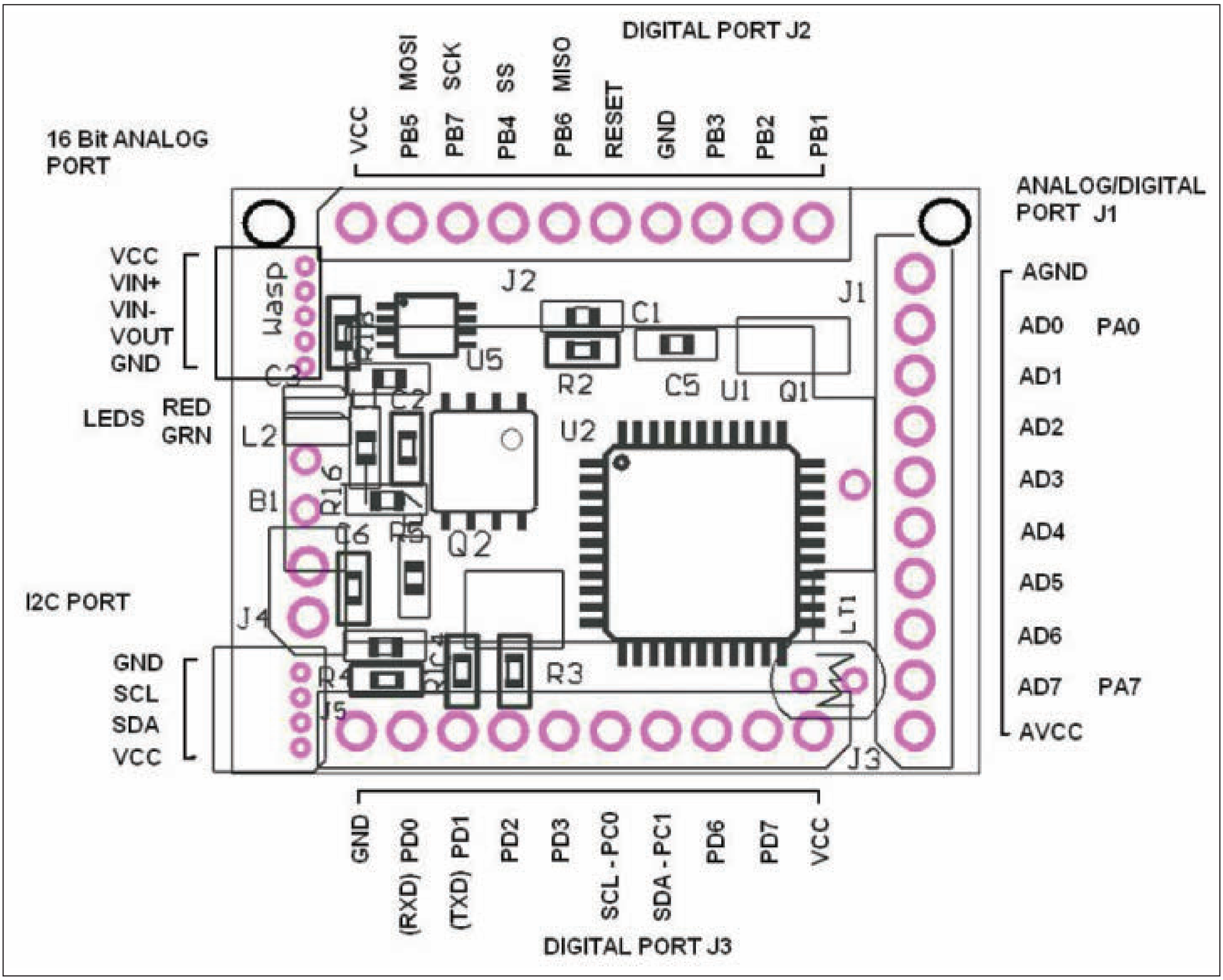


PHOTO 3. Here's the scoop on the AVRISP mkII ICSP connector. All we have to do is match the pinout you see here to the Wasp's machined pin ICSP points in Figure 1.

FIGURE 1. Note that the optional 16-bit analog port and its supporting IC are not installed on our Wasp in Photo 1. Every electrical and logical connection needed by a user to fly the Wasp is mapped out in this figure.



came with a very nice 10-pin ICSP hardware collection which included an ISP10 parallel port programming device and a 10-pin-to-Wasp conversion connector called a CISP. Unfortunately, my laptop doesn't have a parallel port and my AVRISP mkII is looking for a six-pin male connector. So, the trick will be to fabricate an AVRISP mkII-to-Wasp ICSP adapter.

The AVR business end of the AVRISP mkII is captured in **Photo 3**. I dug out my X-ACTO™ hand saw and separated a set of 3 x 3 pins from some double-rowed pin stock. Using that same hand saw, I eliminated three pins from a 10-pin SIP header. The seven-pin SIP header is destined for a rendezvous with the female side of the Wasp's ICSP machined pins. A bit of plated-through perfboard also fell under the hand saw. Throw the perfboard, the six-pin male header, some wirewrap wire, and the seven-pin SIP into an electronic mixer and you end up with something that looks like the contraption under the lens in **Photo 4**. The large black wire pair trailing off with the AVRISP mkII programming cable in **Photo 4** carries the Wasp's 3.3 volt power supply voltage which is sourced from a 3.3 volt wall wart. The objects that make up **Photo 5** appear when our home-brewed AVRISP mkII and its ICSP programming adapter are mated to the Wasp.

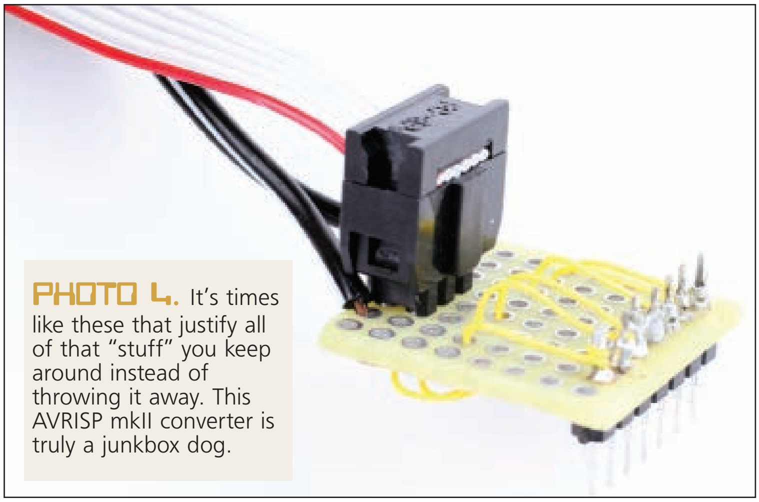


PHOTO 4. It's times like these that justify all of that "stuff" you keep around instead of throwing it away. This AVRISP mkII converter is truly a junkbox dog.

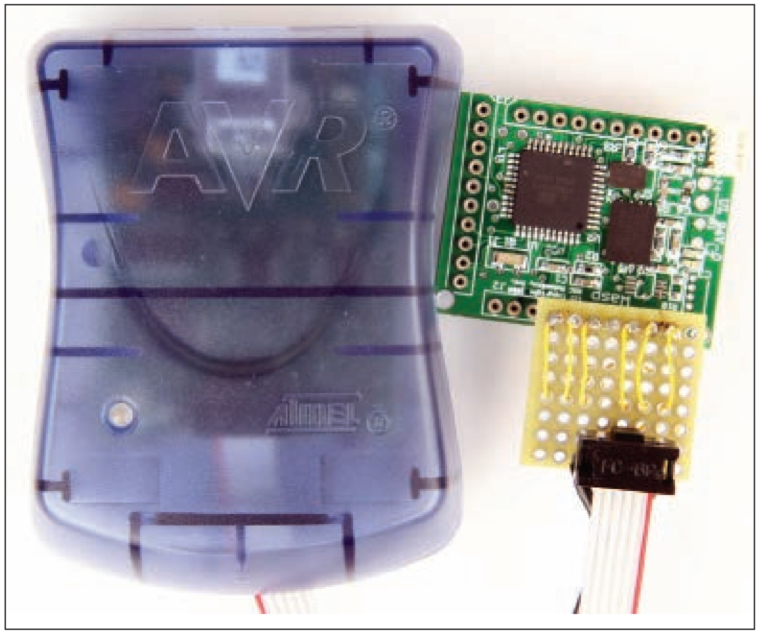


PHOTO 5. One big happy family of AVR stuff ... *and* a junkbox dog.

A Silicon Wasp Nest

If you've had the time and opportunity to follow my article trails in *Nuts & Volts* and *SERVO*, you know that I'm a born-again embedded USB advocate. To my delight, the silicon wasp nest that acts as home base for the Wasp is based on a device called the USB10 Smart AVR USB processor.

The USB10 smart processor — which happens to be the subject of **Photo 6** — is based on an AVR AT90USB162 microcontroller. That means that the nest is just as smart as the silicon insect it serves. The USB10 is equipped with a 16 MHz crystal that moves the CPU bits about at a clock rate of 8 MHz. The reduction in clock speed allows the USB10 to run its internal USB engine with a 3.3 volt or 5.0 volt power supply. The operating power rail is selected via a jumper on the USB10 PCB. The USB10 can only be clocked at the full 16 MHz when the power supply

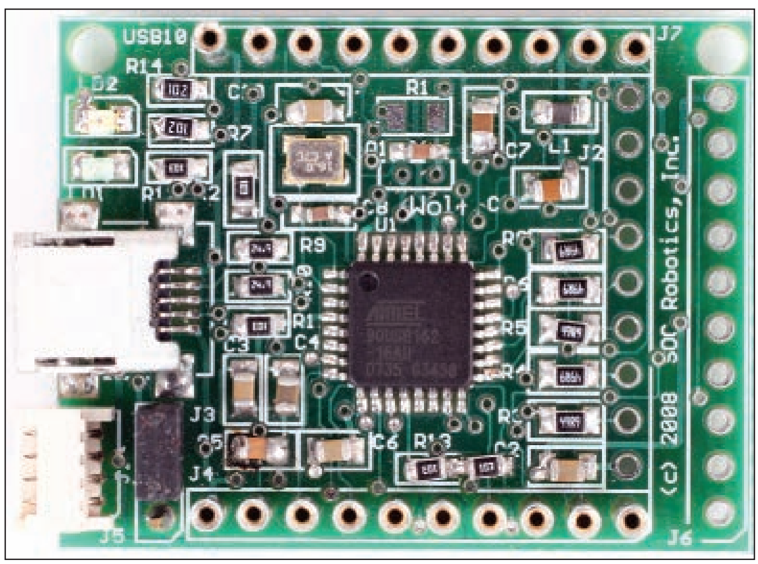


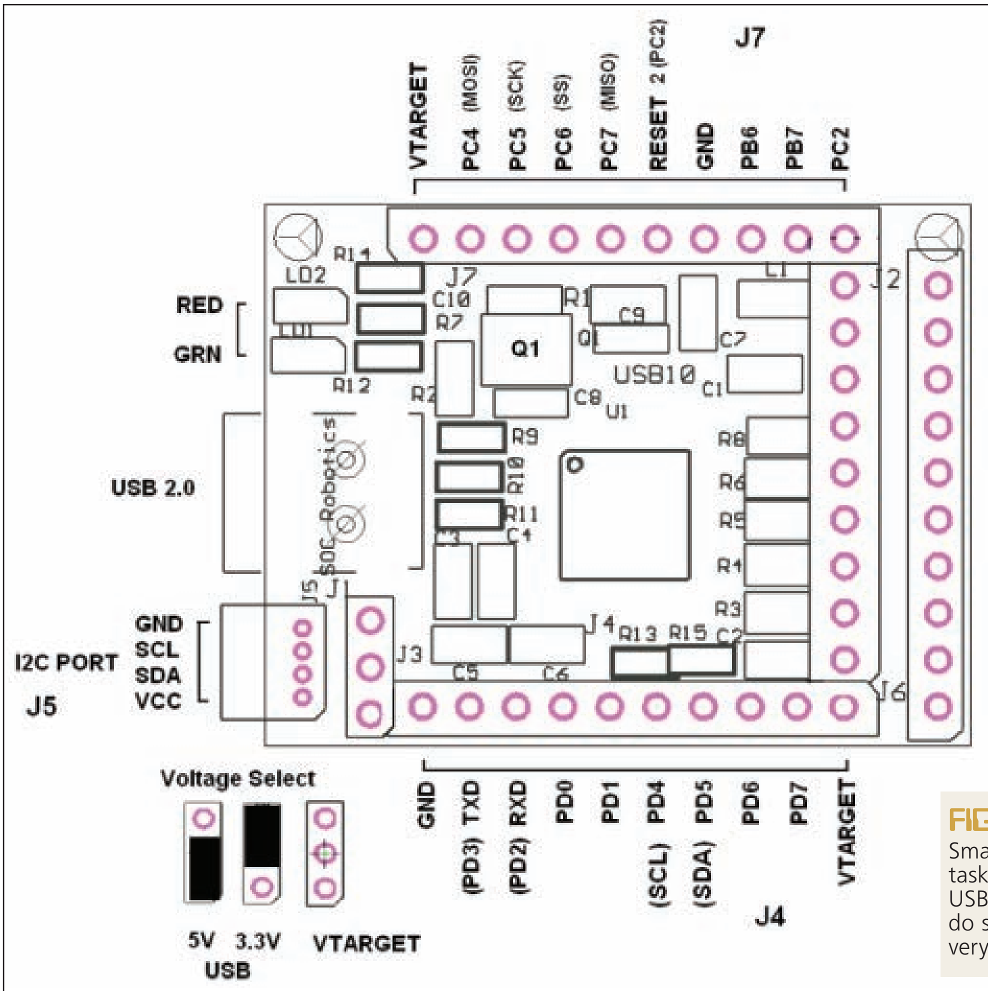
PHOTO 6. This is an aerial view of the USB10 Smart AVR USB processor. The powered TWI portal sits just to the left of the operating voltage jumper.

voltage is 5.0 volts. As one would expect, the USB10 is capable of performing digital I/O operations and is endowed with 16K of program Flash, 512 bytes EEPROM, and 512 bytes of SRAM.

I've never seen a power meter on a nest of a colony of polistes exclamans. However, applying a voltage to the intricate paper art of the polistes exclamans is not at all like putting the juice to a piece of specialized silicon. Although the USB10 is the "nest," it consumes a bit of power due to the AVR that pumps its blood. Like the Wasp it serves, the USB10 consumes power relative to the work it is being asked to do. Current consumption of the USB10 ranges from 8 to 25 mA which is dependent on the clock rate and active internal peripherals.

To get an idea of how the Wasp and its "nest" cooperate, compare the pinouts shown in **Figures 1** and **2**. The first of the good news is that the USB10 can be programmed with our junkbox dog interface coupled with an AVRISP mkII. The ICSP pins on the USB10's J7 associate directly with the Wasp's J2 interface.

Most all of us have been around the microcontroller RS-232 long enough to know that opposing microcontroller UARTs need to feed the other microcontroller's UART receive pin from its transmit pin, and vice versa. Take a look at the USB10's UART transmit and receive lines versus the transmit and receive line positions of the Wasp. The transmit/receive null modem arrangement between the



Alternate Functions J6

JTAG	ISP	IO	MC-USB
GND	GND	GND	GND
5VDC	5VDC	5VDC	5VDC
3.3VDC	3.3VDC	GND	GND
GND	GND	PB5	ESTOP
RST	RST	PB4	LIMIT
TDO	MISO	PB3	SCL
TCK	SS	PB0	SDA
TMS	SCK	PB1	DIR
TDI	MOSI	PB2	STEP
VTARGET			VCC

FIGURE 2. Although the USB10 Smart AVR USB processor is primarily tasked with supporting Wasps, the USB10 processor can stand alone and do some serious data processing of its very own.

USB10 and the Wasp is hardwired in.

I^{2}C is a "hang onto the bus" type of protocol. Thus, there is no null modem arrangement in the I^{2}C world. I^{2}C devices are designed to be clipped onto the I^{2}C bus in a parallel fashion. Unlike RS-232 devices, I^{2}C devices must have an address to participate in the network.

The USB10 I^{2}C function is implemented in software and is an integral part of the Ferret control program which comes loaded in the USB10's Flash. The Ferret control program includes a USB-to-I^{2}C device function and an ISP programmer element. There is also a variant of the USB10 — the MC-USB10 — that supports motor control via the Ferret program. Source code for this is available which allows you — the user — to add additional applications. A USB software CDC application from Atmel along with open source USB code in the MyUSB package are compatible with the USB10. In fact, the Ferret control program is based on the MyUSB package.

The Wasp's interface pins mate directly with the USB10's set of interface pins. The Wasp-to-USB10 interface enables the USB10 to act as a UART and USB communications device for the Wasp. When the USB10 and Wasp are stacked, the USB10 can also be used to program the Wasp.

Ferrets and the Ferret Control Program

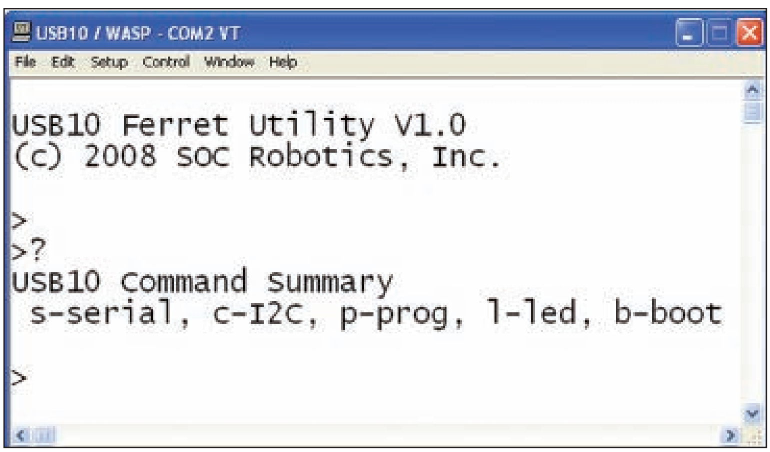
My father became enchanted with ferrets and raised a number of them at our home in Fayetteville, TN. I can still recall my father's first days as a ferret master. We finally figured out why the ferrets would back up to the corners of a room. They were doing their duty, so to speak. That was it. The ferrets lost their "big house" status and were moved to quarters that were previously inhabited by the yard pimps (chickens). Dad wasn't really that angry at them as he outfitted their new digs with lots of pipes for them to explore. Despite the Fayetteville ferrets being ousted from the house, I recall all of them having a distinct personality. There wasn't a one of them that was "born yesterday," if you know what I mean. In its own way, the Ferret control program is aptly named as it is a powerful application crawling around in digital tubes behind a simple user interface.

Opening a terminal emulator session, plugging the USB10 into one of your PC's USB portals, and hitting a key will result in the Signon message captured in **Screenshot 1**. The USB10 is shipped with the Ferret program installed and ready to run. The Signon message is one of five functions contained within the program.

Once I received the Signon message, I entered a "?" at the prompt which revealed the USB10 Command Summary. The first of the five Ferret functions listed in the USB10 Command Summary (s-serial) establish a real time link between the terminal emulator running on the PC and the AT90USB162's UART. In actuality, the AT90USB162's UART is using the USB connection to transfer data to and from a PC virtual serial port (COM1, COM2, etc.). All characters sent to the USB10 from the terminal emulator session are output on the AT90USB162 UART's transmit line. Any characters received by the AT90USB162 UART's receive line are sent down the USB portal to the terminal emulator session running on the computer. Inserting an ESCAPE character (\c) from the PC side will terminate Serial Mode.

The c-I^{2}C command performs a similar task to that of the Serial Mode. However, instead of communicating via the AT90USB162's UART via the USB portal, the communications protocol is transmitted and received on the AT90USB162's USB portal is I^{2}C. The USB10 can act as either a I^{2}C master or I^{2}C slave device. The USB10 default I^{2}C mode is master. Back up and take yet another look at **Figure 1**. You'll count up a total of three I^{2}C interfaces. Actually, you can only use two of them at any time as I^{2}C portals 1 and 2 share their SCL and SDA signal paths.

The p-prog command enables the control program to



SCREENSHOT 1. This Signon message is generated by the Ferret control program that comes factory loaded in the USB10 Smart AVR USB processor's Flash.

use the USB10 SPI portal and digital I/O lines to program the AVR of a nesting Wasp. The USB10 programming function is supported by a PC application called USB10Prog which is a free download from the SOC Robotics website.

What would ANY microcontroller project be without LEDs? As you've probably already figured out, the l-led command allows us to illuminate or extinguish the red and green LEDs that are indigenous to the USB10.

Like the l-led command, b-boot is pretty obvious as to what the function performs. When b-boot is selected, the boot routine loaded in the high memory of the AT90USB162 kicks off. A new program is loaded with the help of an Atmel application called FLIP.

The Ferret control program does all of this and NEVER backs up into a corner.

Where's the Code?

It's already written. The Wasp Data Logger application not only contains data logger routines, it contains separate C modules that enable and exercise the AT90USB162's SPI portal, TWI module, UART, and I^2C functions. You'll also find a C module that initializes all of the Wasp hardware. There's even C code for a monitor and the interrupt handlers. If that's not enough canned C code for you, here's an excerpt from the utilities module that writes data to the Wasp EEPROM:

```
// Function to initiate an EEPROM write
// writes the specified data byte to the
specified location
// this will fail if the EEPROM is not ready!
void RTEEPROMwrite(int location, unsigned char
databyte)
{
unsigned char savedSREG;
EEAR = location;      // set address
EEDR = databyte;      // set data
savedSREG = SREG;     // keep setting so
                      // it can be
                      // restored
CLI();                // disable interrupts
EECR |= BIT(EEMWE);   // set "write enable" bit
EECR |= BIT(EEWE);    // set "write" bit
SREG = savedSREG;     // restore SREG
EEAR = 0;
}
// function to test if the EEPROM is ready
// for a read or write operation
// returns non zero if ready, zero if not
// ready
unsigned char RTEEPROMReady(void)
{
       return !(EECR & 0x02);
}
```

I included the EEPROM ready code to show you that everything you need to interface your C code to a Wasp is most likely already contained in some form or fashion within the data logger application modules. There are even C routines in the data logger modules that can be called to blink the LEDs.

To make the assimilation of the canned Wasp C routines easy, the entire Wasp Data Logger application was written using ImageCraft ICCVX for AVR and Atmel's AVR Studio. AVR Studio is a free download and the ImageCraft C compiler is as inexpensive as C compilers go. ImageCraft also offers a 45 day trial period to allow you to fall in love with their AVR C compiler. SOC Robotics offers a huge bang for the firmware buck, as well. You can download the Wasp Data Logger application, the AT90USB162 Bootloader, the UB10Prog programmer application, and Ferret control program for free from the SOC Robotics website.

I had a blast working with the Wasp and didn't get stung one time. **SV**

Fred Eady can be reached via email at fred@edtp.com

Sources:

ImageCraft
ImageCraft C Compiler for AVR
www.imagecraft.com

SOC Robotics
Wasp
USB10 Smart AVR USB Processor
www.soc-robotics.com

Atmel
AVR Studio
AVRISP mkII
www.atmel.com

HOVERBOT:

Wheels? We Don't Need Wheels Where We're Going

By Paul Verhage

Robots go where you point them and make a dead stop when you command them. Hovercraft, on the other hand, experience far less friction and that makes controlling them conceptually different. Want to simulate a satellite in orbit? A HoverBot lets you do that in two dimensions and in the comfort of your own home. Plus, no flights on the "Vomit Comet" are necessary to test some of your satellite control theories.

Over two years ago, I tried simulating a robotic spacecraft. The attempt involved using the air jets from an air hockey table to lift a Styrofoam sheet into a nearly frictionless state. I was going to have a robot controller and propulsion system sitting on this floating sheet. The robotic spacecraft's propulsion would come from the same compressed air and pneumatic valves some R/C airplanes use to deploy their landing gear.

After discovering that toy air tables were incapable of generating sufficient air flow, I put the idea on hold — but I didn't give up on someday making a hovering robotic spacecraft. Recent Internet searches have uncovered model hovercraft ideas that have inspired me to restart my experiments. The result is the NearSys HoverBot that I'll describe in this and following articles.

Background on the NearSys HoverBot

Hovercraft can travel without friction (or nearly so) because they do away with a robot's greatest source of drag: spinning wheels and motors. Moving through the air at low speed generates such a small amount of drag that a hovercraft gradually slows down when no longer being

It takes three of these EDF-50 ducted fans to make the HoverBot run: one for lift and two for propulsion.

propelled. It's the opposite situation for small wheeled robots; they stop on a dime after cutting power to the drive motors. Low drag conditions mean that driving the HoverBot requires a consideration of Newton's Laws of Motion. The controlling program doesn't plan in terms of distance and heading. Instead, it plans in terms of time and acceleration.

The HoverBot's lift and propulsion comes from three ducted fans (these are GWS EDF-50 ducted fans). The first fan directs air down to lift the HoverBot on its skirt. The other two fans generate a horizontal force that drives and steers the HoverBot. Relays arranged in an H-bridge circuit control the direction the ducted fans spin, and therefore the thrust direction of each fan — except the lift fan. A single relay controls this fan so the HoverBot can take off and land. Perhaps that's a slight misnomer; lift-off is only two inches high.

Even though the force the two horizontal fans generate is small, the HoverBot's low friction lets the fans build up high speeds (in fact, the HoverBot will travel faster than my wheeled CheapBots). The drive fans attach to the HoverBot off-center. This allows the two fans to drive and steer the HoverBot like little rocket engines on a satellite. When fired in the same direction, the HoverBot accelerates either forward or backwards.

Recall your high school physics class that accelerating backwards doesn't necessarily mean the HoverBot moves backwards. If the HoverBot is initially moving forward, the backwards acceleration will first slow it down and bring it to a halt before it even begins to move (displace) it backwards. How long the fans have to drive the HoverBot backwards to reverse its travel depends on how long the HoverBot was accelerating forward.

Firing one fan forward and the other fan backward spins the HoverBot around its center. The longer the fans spin an initially stationary HoverBot, the faster it spins and the longer the fans must work to stop the rotation. I suspect it's possible to spin the HoverBot long enough for centripetal force to tear it apart. Therefore, it's important not to overdrive the HoverBot in a turn. It's best to use a small burst to begin its rotation and then wait awhile for it to slowly approach its desired final heading before reversing thrust.

Parts List

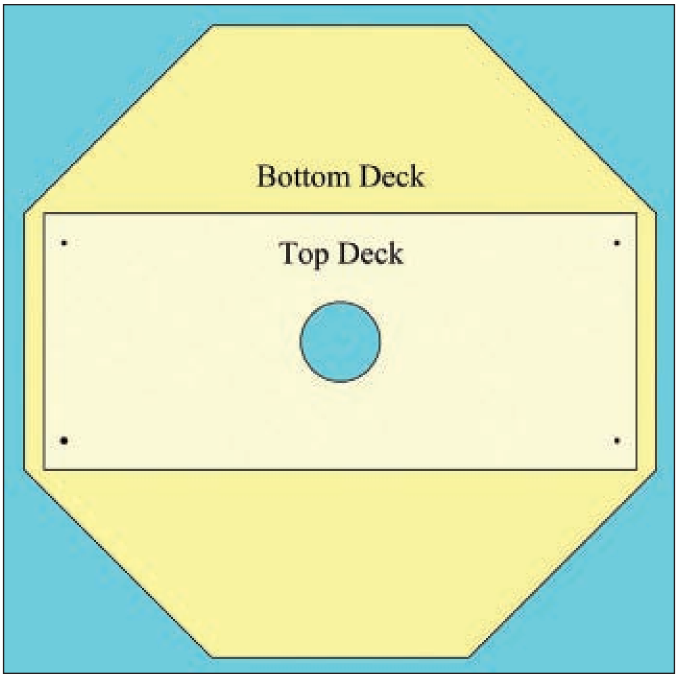
QTY	DESCRIPTION
4	#6-32 bolts, 2 inches long
4	#6-32 nylocks
1	Roll of two inch wide Scotch mailing and storage tape
2	#4-40 bolts, 3/4 inch long
2	#4-40 nylocks
3	GWS EDF-50 ducted fans

- Correplast (corrugated plastic) 5/32 inches thick, available from either a plastics distributor or sign company.
- 0.7 mil plastic drop cloth, available from hardware and paint stores.
- 1/4 inch diameter plastic tube (Evergreen plastic #228), available at many hobby and model train stores.
- 20 mil thick styrene sheet, available at many hobby and model train stores.
- Sheet of 10 mm thick Cellfoam-88, available at craft stores (note that 1/2 inch thick Styrofoam sheets from home improvement stores also works).

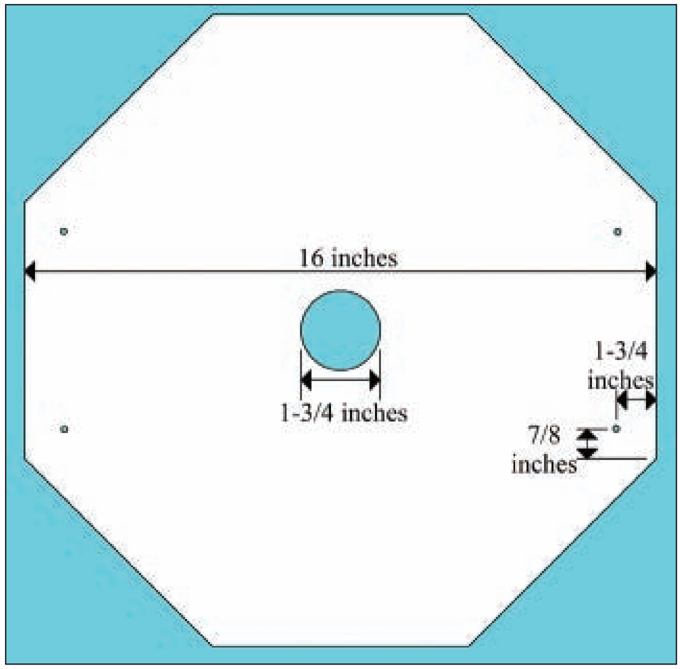
Building a HoverBot

In writing this article, I realized how involved the HoverBot has been, so my article will be broken into a few parts. This month, we'll cover the construction of the HoverBot body sans its skirt. Next time, we'll build a skirt and add the electronics. So to begin with, collect the materials listed in the **Parts List** to build the HoverBot.

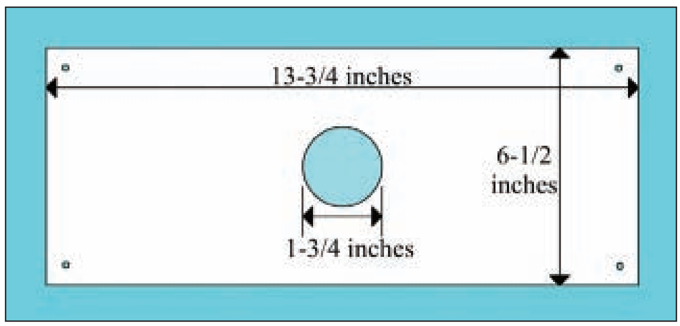
The plan is to build two decks: one for the top and one for the bottom. The bottom deck houses the lift fan, nozzle, and skirt. The top deck houses the batteries, robot controller, fan electronics, and drive fans. Separating the two decks are bolts and spacers. We'll need to align all the holes in both decks and prepare the bottom deck for the fans and skirt.



Center the top deck as perfectly as possible, but never let the fan holes get out of alignment. It's more critical that the fan holes line up properly than the decks.



Here are the dimensions for my HoverBot's bottom deck. You can make yours larger or smaller. However, a larger HoverBot may not lift as well.



Note that the top deck is not as wide as the bottom deck.

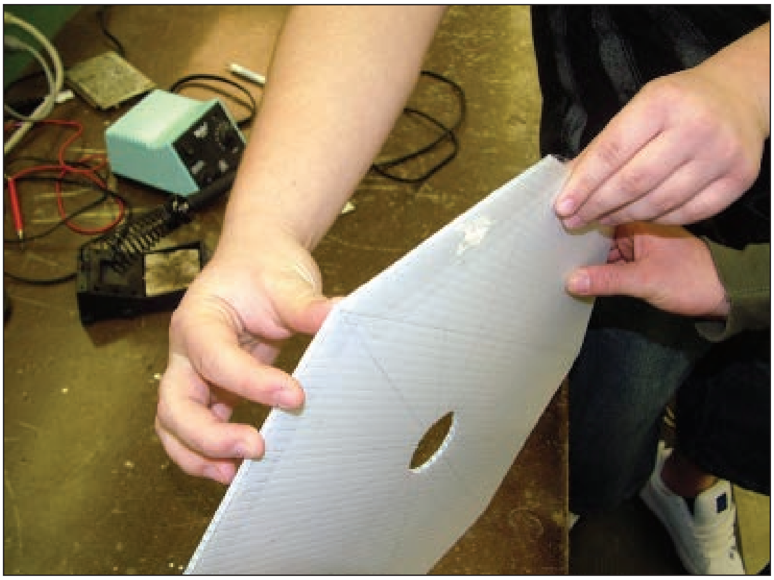
Bottom Deck

It's best to cut out and drill the holes in both the top and bottom decks before moving on to complete the bottom deck. That way, the holes in both decks align with each other. So, begin by cutting two pieces of 5/32 inch thick Correplast to the shapes and sizes illustrated below. A utility knife and metal straight edge work best for this. Use a T-square to lay out the lines and ensure the HoverBot remains square. Don't worry about the center or four corner holes at this time. Just mark their location.

You can locate the center of each deck by drawing diagonal lines from the corners. Alternatively, you could use a ruler and measure to the center. After marking the center of each deck, use a T-square to draw right angle lines that intersect the center of each deck. Place a 1-3/4 inch circle template at the center point of each deck and draw the circle on the Correplast. This hole marks where the fan will blow air to lift the HoverBot. Use a small-bladed Exacto™ knife and carefully cut out the fan holes. Next, stack the top deck on the bottom deck and align the fan holes. The stacked decks should look like this.

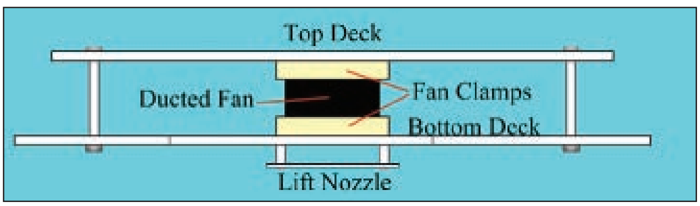
There are four holes in the corners of the decks for the riser bolts. These holes must line up if the lift fan is to sit vertically in the HoverBot. Therefore, after aligning the top and bottom decks use several pieces of masking tape to keep them aligned while you drill the riser holes. I find a Dremel is the easiest way to make these holes.

The risers are #6-32 bolts, so a 1/8 inch drill bit will suffice to make these holes. The exact positions of the riser holes are not critical, but keep them at least 1-1/2 inches from the edge of the bottom deck. After drilling the first



It's easier if you have an extra set of hands. My assistants are applying clear mailing tape around the open edge of the bottom deck. The sticky side of the tape holds it firmly to the Correplast.

hole, you may want to place a bolt through it to help keep the decks in alignment while you drill the rest. After drilling the riser holes, separate the decks and set the top one aside.



The lift fan is easily removed from the HoverBot because of the Styrofoam bands holding it in place.

Note the bottom of the top deck with its fan clamp and the ducted fan that it's holding in place.

The skirt tapes to the bottom deck, but the edge of the Correplast is sharp enough to cut the plastic skirt, so we're going to cover the edges before proceeding any further. Use two inch wide mailing tape as a bumper; apply it over the open edge of the deck and fold it over as illustrated below.

Lift Fan Clamps

Now that the top and bottom deck are cut out, it's time to think about mounting the lift fan. Rather than hot-glue the lift fan between the HoverBot's top and bottom decks, we'll use two Styrofoam rings. Each ring is glued to its deck so when the decks come together, the lift fan is trapped firmly within the clamps.

Cut two squares of 10 mm thick Cellfoam 88. The squares are three inches on a side and have holes 2-1/8 inches in diameter cut in their middle. Glue the first fan clamp to the bottom face of the top deck and the other to the top face of the bottom deck.

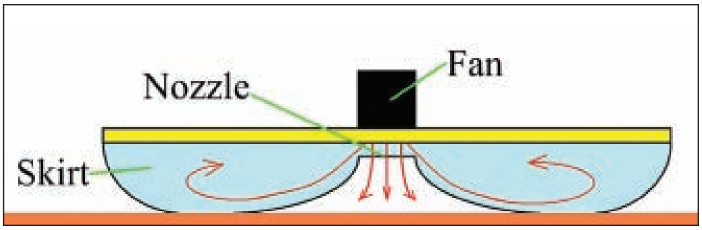
Lift Nozzle

Air from the lift fan must fill a flexible skirt to create the HoverBot's cushion of air. Air pressure inside the skirt attempts to keep the skirt in contact with the ground and only a thin stream of air jetting evenly out of the sides of the skirt lifts the HoverBot. For the skirt to do its job, the outside edges of the skirt must be lower than the center as illustrated below.

The material for the HoverBot skirt is 0.7 mil-thick plastic drop cloth. A bag of drop cloth has enough material for a fleet of HoverBots and is available at most paint and hardware stores and many big box retailers. Don't use the 1.0 mil or thicker drop cloth since this isn't as flexible and doesn't conform as well to variations in the floor.

For the skirt to do its job, the center of the bag must be closer to the lift fan while still allowing air from the fan to flow into the skirt to inflate it. Some toy hovercrafts keep the center of the skirt close to the lift fan with strips of clear plastic tape. Rather than use tape, I built a plastic nozzle for the HoverBot skirt. Here's how to build this simple, lightweight nozzle.

Draw two octagon shapes on a 20 thousandths of an inch thick polystyrene plastic sheet that measures three inches across. In the center of these octagons, draw a



Since the floor has variations in height, the HoverBot's skirt must be flexible to maintain its cushion of air. If the skirt wasn't flexible, the air cushion would blow out one side of the HoverBot causing it to crash to the ground. A plastic bag attached to the bottom deck and filled with air from the lift fan is perfect for doing the job.

After gluing the fan clamp on the bottom deck, use a sharp Exacto knife and cut out two notches 1/2 inch wide. The nylocks from the nozzle will attach here. Notice that the hole in the deck where the fan draws air is slightly smaller than the diameter of the hole in the fan clamp.

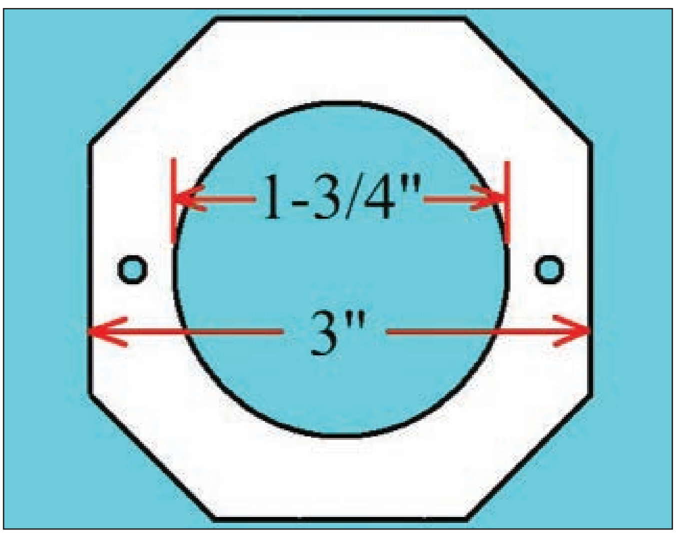
1-3/4 inch diameter circle. Now, cut out the octagons and their center circles with an Exacto knife or scissors. Next, drill two holes — one centered on each side — in the nozzle plates with a 1/16 inch drill bit. I recommend stacking the nozzles when drilling the holes so they align properly.

Two bolts and spacers hold the nozzle plates the proper distance from the bottom deck. I found a good spacer to be two #4-40 bolts 3/4 inches long, two #4-40 nylocks (nylon lock nuts), and two 1/4 inch diameter styrene plastic tubes cut 3/8 inches long.

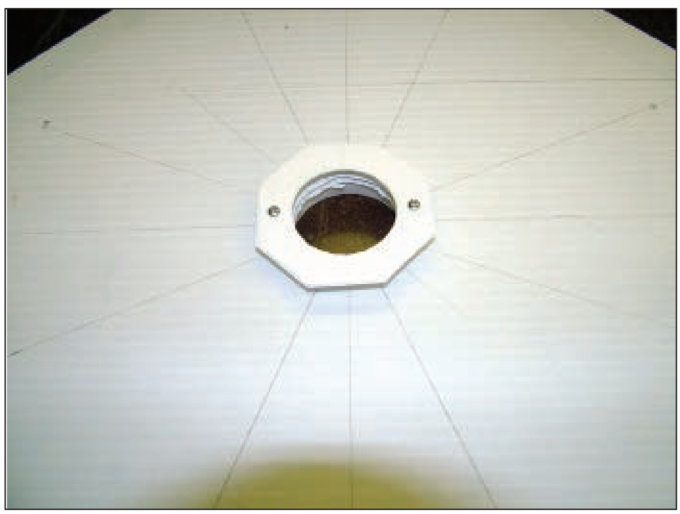
To ensure proper alignment of the nozzle, center one of the nozzle plates over the lift fan opening in the bottom deck and mark the location of the two spacers. Then, drill two 1/8 inch holes in the bottom deck for the spacers.

Now, test the nozzle by stacking the two nozzle plates, running the #4-40 bolts through them, stacking the plastic tubes over the bolts, and then attaching the nozzle plates to the bottom of the bottom deck. The nozzle will look like the image below this.

Next month, we'll start on the skirt and electronics. NearSys is about to produce a HoverBot kit, however, my articles will contain enough information that a *SERVO* reader can build one from scratch. (Just don't start selling NearSys HoverBots!) Check out the NearSys YouTube channel to watch a short video on the HoverBot. I think you'll be impressed enough with its performance that you'll want to make one for yourself. Be prepared for a new paradigm in robot control. **SV**



The HoverBot requires two of these nozzle plates.



The nozzle plates are held 3/8 inches below the bottom deck. In this image, the bottom deck is upside down.

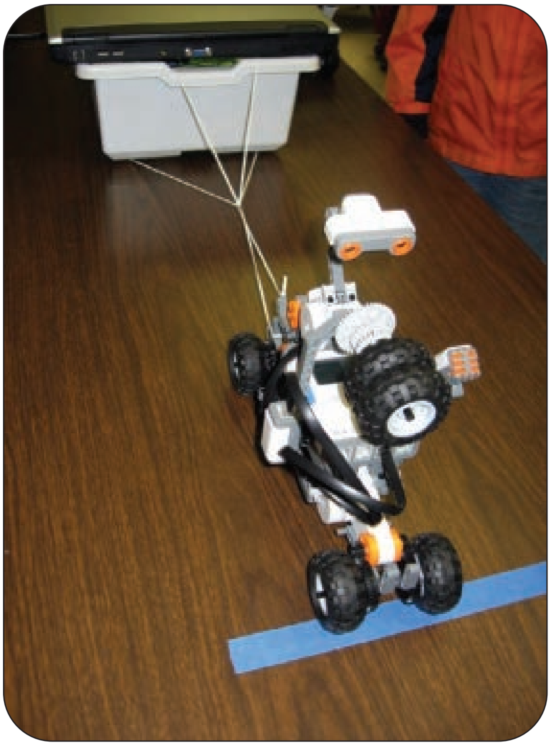
MENAGERIE

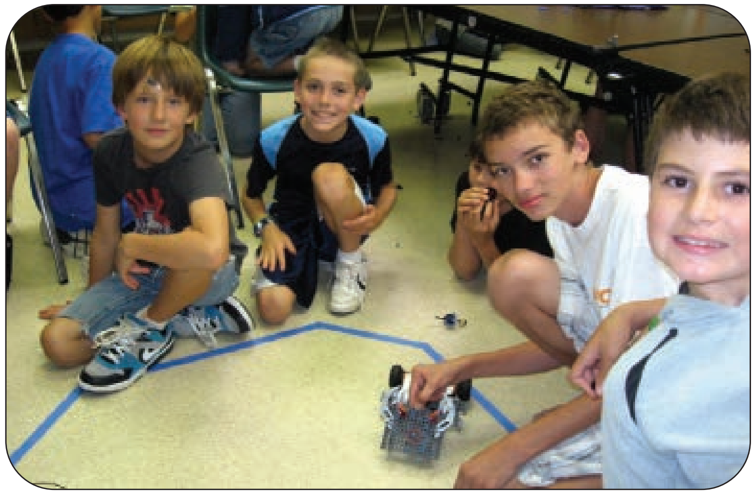
LEGO Robotics Camp

There's nothing that can compare to the excitement of kids working with LEGO robots, and the valuable lessons they take with them from the experience — in not only engineering and programming, but teamwork as well. Freelance writer Greg Intermaggio ran a one week LEGO Robotics camp this summer in San Rafael, CA. Kids worked on an average of five unique robots throughout the week, learning basic programming concepts, as well as mechanical engineering concepts like gear ratios, friction, and simple machines.

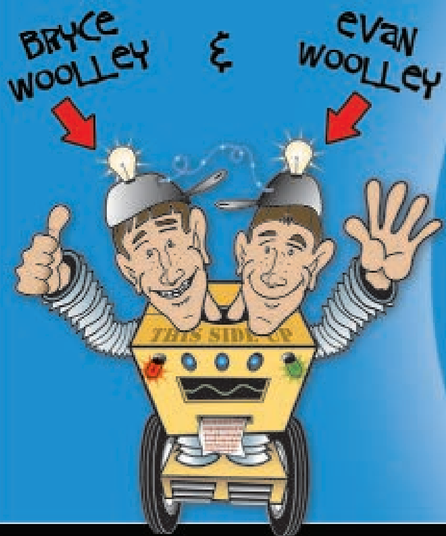
By the end of the week, students had seen a progression of challenges — from building a basic driving chassis that can drive in a straight line to creating a sumo robot geared down for torque and designed to push others out of a ring.

If you live in the Bay Area and have a kid interested in LEGO Robotics, contact Greg at G.Intermaggio@gmail.com.





Twin Tweaks



THIS MONTH:

13 Years Under The Sea

SSC TRANSDEC, SITUATED ON PICTURESQUE POINT LOMA.

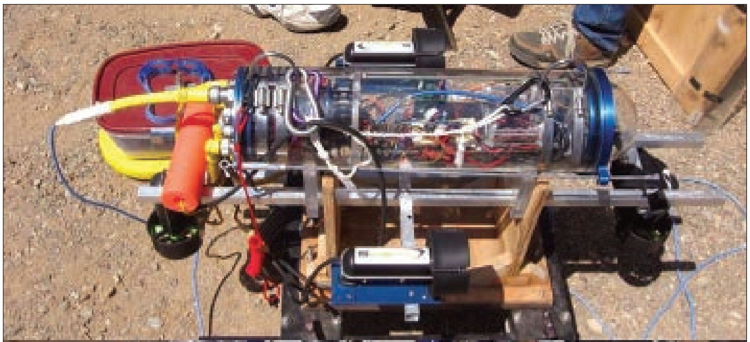
We have a special place in our hearts for underwater robots. In the August and September '08 issues of *SERVO*, we detailed our aquatic odyssey as part of the 2008 MATE ROV Competition where a kit from !nventivity propelled us and the rest of Tau Beta Pi California Ψ to awards in team spirit and excellence in our engineering evaluation. A lustrum ago, we were amazed by the ingenious teams that tackled the incredibly difficult challenge posed by the 8th Annual AUVSI and ONR International Autonomous Underwater Vehicle Competition (see the November '05 Issue for the full story).

In 2010, the Association for Unmanned Vehicle Systems International (AUVSI) and the Office for Naval Research (ONR) held their 13th annual International Autonomous Underwater Vehicle (AUV) Competition (now rebranded as the RoboSub Competition). After five years, we thought another visit was appropriate, especially given that AUVs and their cousins (ROVs) seem to be getting more press about their versatile problem-solving ability (check out the July '10 issue for a good example!). The competition was held from July 13th-18th at the SPAWAR Systems Center San Diego Transducer Evaluation (TRANSDEC) facility in California.

Sign of the Sea Bot

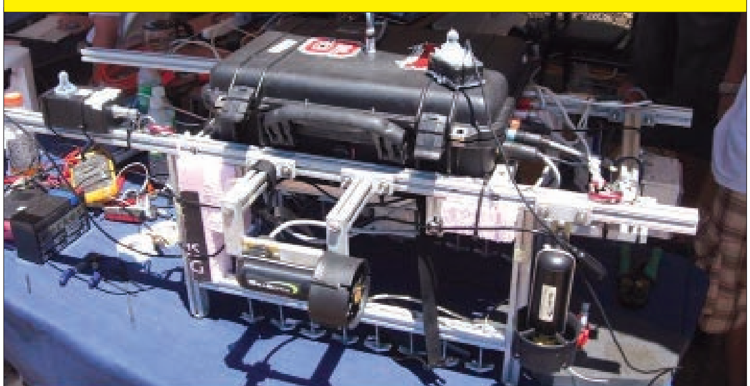
Upon arriving in sunny San Diego, we got a chance to talk with Daryl Davidson, the Executive Director of the AUVSI Foundation. AUVSI and the AUVSI Foundation are separate organizations that are fundamentally linked. AUVSI is a 501(c)(6) trade organization chiefly concerned with business to business activities that advance the technology of unmanned systems. The AUVSI Foundation is a 501(c)(3) organization dedicated to fundraising and investing in education. While AUVSI focuses on the here and now, the

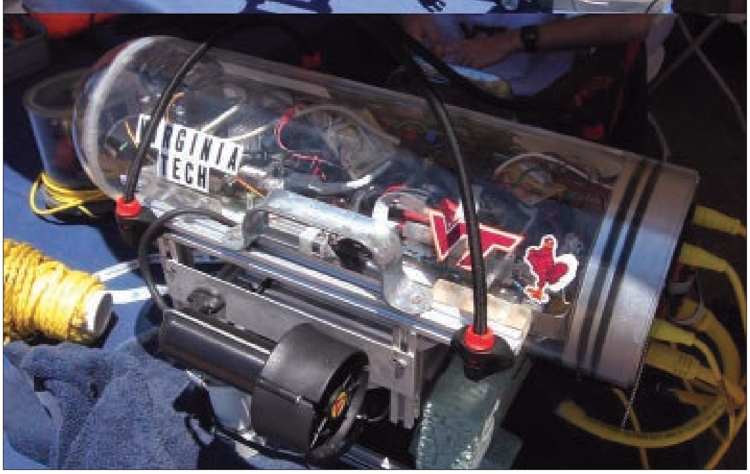
AUVS HAVE TO NEGOTIATE HEDGES, COLLECT LIFE VESTS, AND CHOOSE THEIR WEAPONS WISELY IF THEY WANT TO SURVIVE!



Veteran competitors Amador Valley High School (top) and University of Maryland (bottom) brought large crews to sunny San Diego.

North Carolina State University (top) and Virginia Tech (bottom) wanted to make a strong showing for the South.





Foundation is all about the future.

The Foundation is deservedly proud of the legacy that it has already established in 13 years. Alumni have been handpicked by organizations like SPAWAR to take advantage of the formidable skill set they develop working on AUVs.

Speaking of that formidable skill set, we were amazed by the leap forward in difficulty of the tasks this year compared with previous years. Five years ago, the course was based on a scenario of pipeline repair. Teams were tasked with dropping a marker on the orange colored section of the black pipe that signified damage. That year, everyone had trouble with the vision based pipe repair (an eerie premonition). Now, almost every task has a very difficult vision component. Davidson explained that as computing power has leapt forward in recent years, so has the ability of the teams to create sophisticated systems for less cash. AUVSI is constantly resetting the bar to keep it high enough for the dedicated veterans. The difficulty, however, is to keep the courses challenging enough for veteran teams while still keeping the competition accessible to rookie teams.

Despite the intensely difficult task of building an autonomous submarine, all of the teams have a great attitude about the competition and they realize it's not all about bragging rights. Many teams have the simple goal of beating Dave's course (Dr. Dave Novick, Technical Director of the Foundation).

Even after spending mere minutes there, we were really impressed by the collegial atmosphere of the competition. Teams were always ready to help with an extra tool or troubleshooting advice. Davidson says that the Foundation would love to take credit for the cooperative atmosphere, but all the credit belongs to the students. We suppose that such an organic growth of collegiality seems like a natural consequence of the comprehensive program that AUVSI has created.

In addition to building the robots, teams have to make engineering presentations, write journal articles, create team websites, and reach out to companies to form a sponsor network. AUVSI really captures every aspect of the professional engineering process, and that process demands teamwork. Not only do these tasks facilitate the development of well rounded engineers, they also provide a way for rookie teams perhaps intimidated by the difficult technical challenge to still be competitive. Teams that need a few years to sort out machine vision can still create professional level journal articles and engaging technical presentations. Davidson appropriately refers to the AUV Competition as a "small business incubator" and after meeting the teams, we couldn't agree more.

Deep Blue Pool

This year's tasks were horrifyingly difficult and inspired by the horror classic *Friday the 13th*. We

suppose it seemed a bit more whimsical than a game inspired by the massacre of Knights Templar, but its appropriateness became clear when we learned that movie themes have been the norm since the 10th Annual AUV Competition. The Roman numeral for 10 inspired a Pirates of the Caribbean theme, followed by Ocean's Eleven, the Diving Dozen, and now Underwater the 13th. Next year's theme is anybody's guess!

Escaping a psychotic movie killer must be good motivation to write a nice elegant program for your autonomous sub. As always, the first unofficial task of the competition is to swim through a validation gate. AUVs then follow a path made out of bright orange PVC, picking up points for each section of the path that's completed.

The first official task of the game is to collect life vests (we're talking about Camp Crystal Lake, after all). The vests are represented by three buoys submerged at different depths. The buoys are green, yellow, and red, and AUVs are tasked with hitting two buoys of a specific color in a specific order.

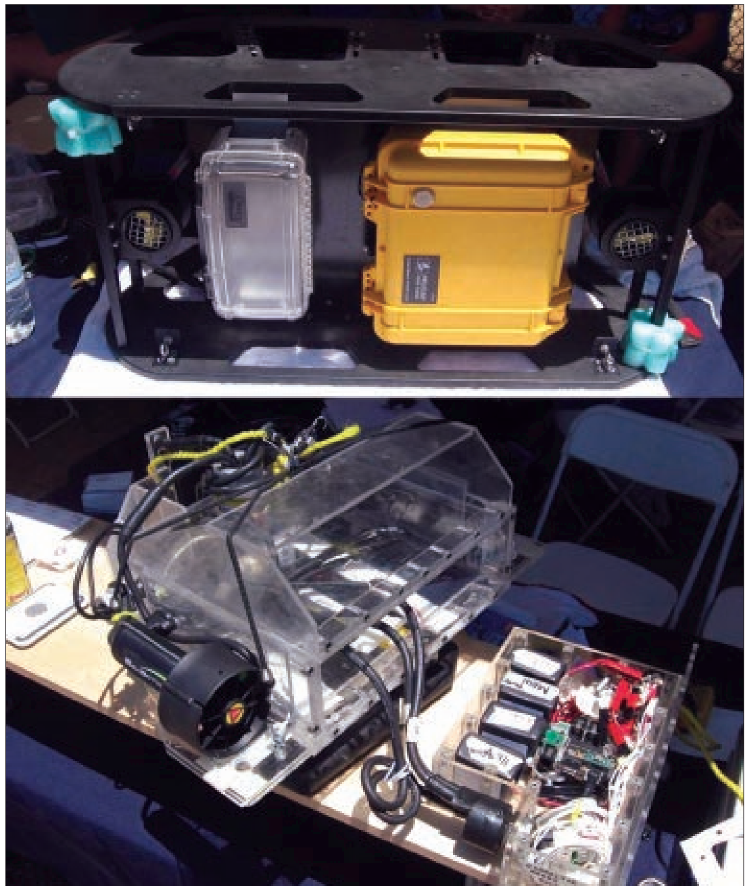
The next task is to leap a "hedge" made out of green PVC a bit nearer to the surface. Clearing the hedge will give AUVs the chance to pick up weapons to defend themselves (figuratively, of course). A PVC structure is located at the bottom of the pool with four bins. The bins contain outlines of hedge clippers, an axe, a hammer, and a machete. The AUV must drop a marker in a previously selected primary weapon bin and secondary weapon bin for maximum points. The other bins count as weapons of choice and will still garner the teams some points.

The AUVs aren't completely defenseless because the next task is to shoot a torpedo through a small 18 x 18 inch window. There are four windows to choose from, and the sub must shoot through the properly colored target (red, green, yellow, or blue).

The final tasks are a bit more altruistic. AUVs must surface in a PVC octagon (representing a police station) to complete their run. There are two octagons to choose from, with the proper one identified by an acoustic pinger. To score big, AUVs can bring a "camp counselor" with them into the police station. The counselor is a vaguely humanoid PVC skeleton that had the good sense to position itself below the police station before being incapacitated by Jason.

Teams have to do all of this in 20 minutes which includes five minutes of preparation time. If teams complete a run and have time left over, they can choose to make another run. If they do, the score from the second run is the one counted — even if it is worse than the first.

A unique aspect of the AUV Competition is the rule set. Many robotics competitions include comprehensive sets of rules meant to close loopholes and ensure that all teams meet on a completely level playing field. The

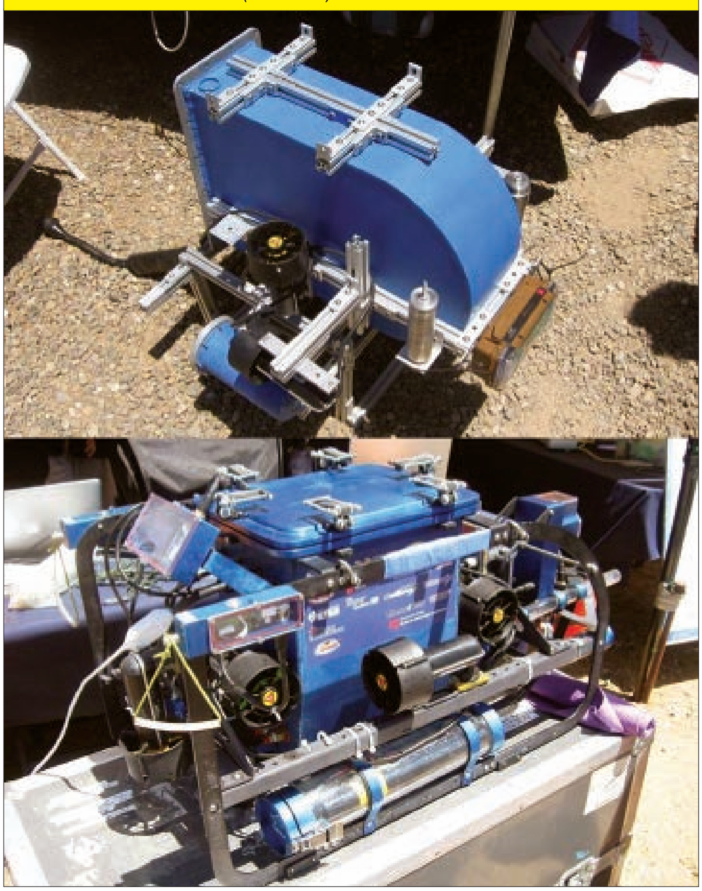


San Diego City Robotics and Pacific Nautilus hoped to make the most of the home field advantage.

The Kyushu Institute of Technology from Japan (top) and KAIST from South Korea (bottom) showed that you can make a long trip in style.

SAN DIEGO IBOTICS (TOP) AND UC IRVINE (BOTTOM) MADE THE UNIVERSITY OF CALIFORNIA LOOK GOOD WITH CARBON FIBER AND ALUMINUM.

THE AUTONOMOUS ROBOTIC VEHICLE PROJECT'S BEARACUDA (TOP) AND ETS'S SONIA (BOTTOM) AIMED TO MAKE CANADA PROUD.



AUV Competition rule set is positively spartan by comparison. Other than some size (6′ x 3′ x 3′), weight (110 lbs max), and safety (torpedoes must not go fast enough to bruise someone) regulations, there are very few limitations on creativity. As Davidson so eloquently put it, more draconian rules would be "otherwise squelching innovation."

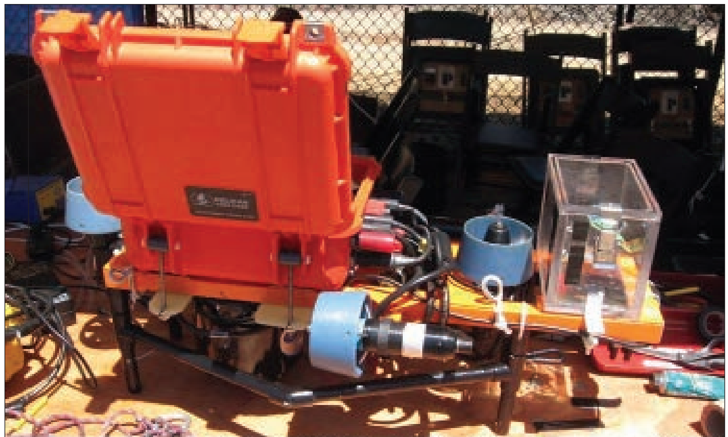
Living the Life Aquatic

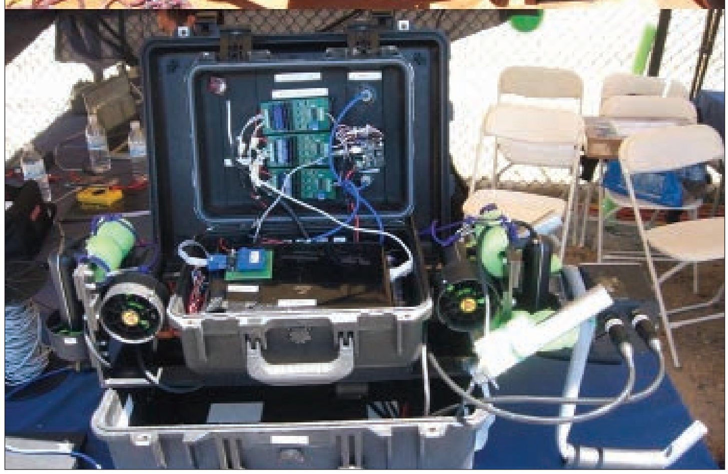
After becoming reacquainted with AUVSI and brushing up on this year's game, we were finally ready to meet the intrepid teams. One of the first teams we talked to were defending champions Cornell. Cornell had a fantastic run last year. On their first try, they completed every task, and with time left over they even got bonus points. In response, AUVSI has raised the bar by including even more devilishly difficult vision based tasks. Despite such past success, the Cornell team refuses to rest on their laurels. The team is composed entirely of undergrads and is completely student run. Their primary goal is to train students, and their comprehensive organization includes business, PR, and community service elements. On a technical note, this year they have been working hard to deal with illuminant metameric failure — the problem that things look different under different lighting conditions.

Another veteran competitor was the University of Texas at Dallas. UTD had something to prove this year after seven years of competition that has seen them as high as second place two years ago, but drop to fifth last year. UTD's 2010 AUV sports a modular design that they think might make them the frontrunners for the Best Kludged AUV Award (if there was such a thing). Kludged or not, the AUV looks formidable, and in the words of their team mentor, they've done "pretty well for a landlocked school."

Robotics @ Maryland from the University of Maryland were awarded the Most Promising Rookie award four years ago when they placed tenth. In their second year of competition, they placed first and last year, they placed eighth. To prove that the early vote of confidence was not made in error, the team this year has created a purely functional design meant to complete the tasks as effectively as possible. The Maryland AUV is the only robotic sub implementing satellite control theory, and the Maryland team is the only one with multiple team t-shirts for different days of competition. The murderous Jason turtle on one design gave the impression that their AUV (the Tortuga III) would be a positively dangerous competitor.

Another past winner was the University of Florida, who won last time we visited in 2005. This year, their AUV was still named the SubjuGator, but it was an ambitious new design. Such ambition seems fitting for such a storied





THE US NAVAL ACADEMY (TOP) AND EMBRY-RIDDLE AERONAUTICAL UNIVERSITY (BOTTOM) MADE GREAT USE OF WATERPROOF PELICAN CASES.

team — the University of Florida has participated in every single AUV Competition to date! The state of Florida is also the home of the team from the University of Central Florida who — with Lovecraftian flair — named their sub Cthulu.

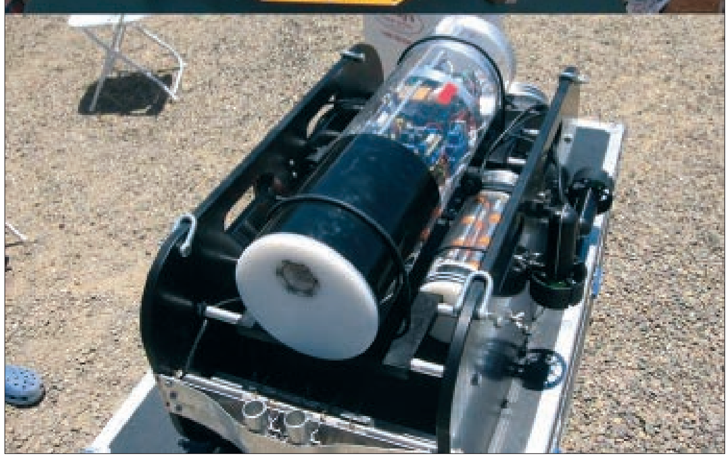
When we stopped by the work area of veteran competitor USC, the team was hard at work sorting out battery issues for their polished looking AUV named Seabee III. And while the batteries may have been troublesome, USC had no trouble with their implementation of ROS, the Robot Operating System base from developer Willow Garage.

We were very impressed by how many teams hoped to tackle all of the vision based tasks instead of focusing on one or two and calling it a day. The Virginia Tech sub was designed to complete every task, and a modular design allowed the team to detach and work on systems that were being troublesome (when we stopped by, they were sorting out the sonar). VT's neighbors to the south — North Carolina State University — were also their neighbors in the competition, with their pit area in the next booth over. Veteran team NCSU was working on a tight budget this year, with the ingenuity and efficiency of their team summed up with their custom waterproof switch that consisted of a bottle nipple and a generous amount of silicone.

We imagine that it would be very easy for successful veteran teams to stick with the same basic sub design, simply tweaking it from year to year. This is often not the case, and teams will often go through dramatic redesigns from year to year. The team from the US Naval Academy did a fundamental redesign of their sub two years ago, and believed that a Doppler Velocity Logger and cameras with auto white balance could put them in the finals.

In addition to the teams coming from the far corners of the United States, sunny San Diego was home to numerous teams. San Diego City Robotics is a team made up of students from San Diego City College and SDSU. When we stopped by, they were working away on some problems that arose during testing, though with a sunny demeanor fitting of their hometown. They looked at the problem solving as "part of the fun, not part of the stress."

San Diego iBotics is a non-profit student run organization that draws most of its team from UC San Diego. Their AUV — the Stingray — is truly a thing a beauty. Its carbon fiber shell looks like its oceanic namesake, and we were surprised to learn that the team



TEAMS CAME FROM AS NEAR AS USC IN LOS ANGELES (TOP) AND AS FAR AS ICELAND — AS WITH THE UNIVERSITY OF REYKJAVIK (BOTTOM).

Veteran competitors like the University of Texas, Dallas (top) and the University of Florida (bottom) strive to build on past success.

designed the sub from the top down — starting with the hull design and then working to fit everything inside. Perhaps their artistic flair is what pushed the team to get creative with propulsion. In addition to three traditional vertical thrusters, the Stingray is outfitted with two Voith-Schneider propellers that allow for any possible movement in the horizontal plane.

One team that stood out for their high aspirations was Pacific Nautilus. Pacific Nautilus is actually a student run non-profit group made up of community college students. Now in their fourth year of competition, Pacific Nautilus hopes to use the brain of their AUV for both the underwater and ground competitions next year. Eventually, they hope to use the same unit for the aerial competition, as well.

The team from Amador Valley High School was the only high school team in the competition. AVHS is a fierce competitor, having competed since 2000 and taking second place in 2001. The team has perfected the process of recruiting and retaining members, and had one of the largest teams at the competition. Without tons of resources, AVHS focuses on improving their sub from year to year. Having stabilized their electronics, this year they were testing out a new Linux based Beagle Board.

Joining the Sea Hunt

The 2010 Competition also welcomed a group of newcomers unfazed by the extreme difficulty of the technical challenge. First and second time competitors came from as far away as Montreal from McGill University, or as close as Southern California as with UCI. The rookie UCI team dealt with the challenge of preparing for competition on a shoestring budget, so it's a good thing they had some management students to keep things running smoothly. In the words of the team, their AUV was "built with a handsaw, a tig welder, and a couple of files." You wouldn't be able to tell though, by looking at their sleek aluminum sub. With Matlab written algorithms, the team aimed to tackle the vision tasks even in their first year, and they were quick to thank UCI for all of their support — particularly in the form of their own lab.

Sophomore competitor Embry-Riddle Aeronautical University used some of the popular waterproof Pelican cases in their design, and were eager to put their LabVIEW-powered sub to the test after some trouble with robot transportation last year.

Cornell's AUV Tachyon makes a practice run.

One Ocean Y'all

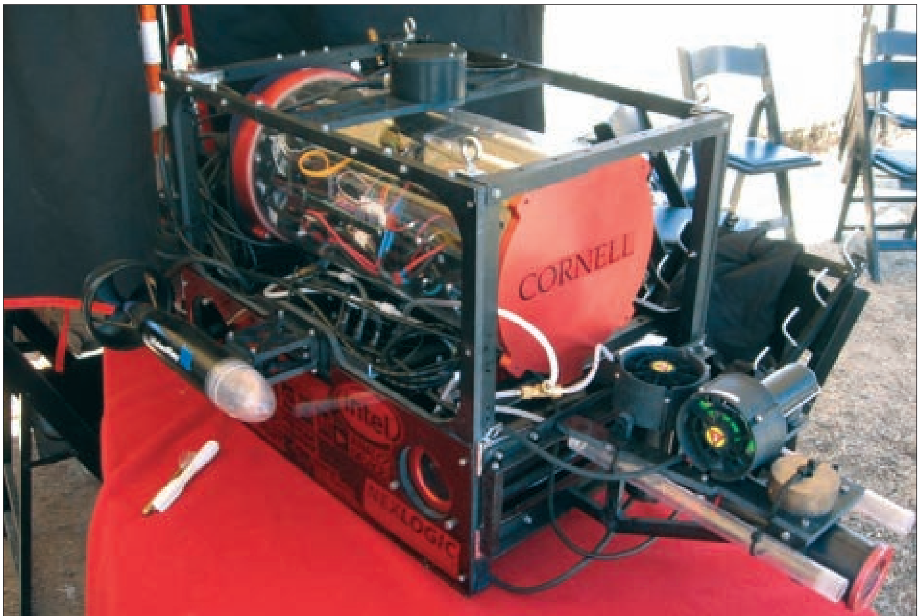
Putting the "International" in AUVSI, several teams made long treks from all over the world to participate in the 13th Annual

SPECIAL THANKS TO

Daryl Davidson

Robosub Competition. Another rookie team came all the way from Iceland. The University of Reykjavik had been doing well with their qualifying runs by the time we talked to them, but they were sorting out leaks in preparation for the next rounds. The Reykjavik team said one of their biggest challenges has been dealing with lighting conditions, because apparently things in San Diego look a bit brighter than back in cloudy Iceland.

The Kyushu Institute of Technology in Japan also sponsored a team. The veteran Kyushu team redesigned their sub to be smaller this year, and the team is known for always turning in excellent technical reports.



PRACTICE MAKES PERFECT. CORNELL MAKES IT A REPEAT!

KAIST Bio Information Systems Lab from South Korea sponsored a team that produced a truly stylish AUV that looked like a turtle. To prove that you can maintain a great sense of humor even when dealing with mind-boggling programming challenges, KAIST designed their required AUV kill switch to be activated by pulling off the turtle's head.

Two teams hailed from Canada. The Autonomous Robotic Vehicle Project — based at the University of Alberta — felt the weight of the entire pacific northwest of North America on their shoulders. Their compact sub — the Bearacuda — seemed to handle the pressure nicely with solid mechanical systems.

Team SONIA from École de Technologie Supérieure in Quebec has participated since 1999. They used the same hull as last year but with new electronics. In addition to solid AUVs, ETS had one of the best team websites, and they promised to upload video logs from their runs to give visitors a robot's eye view of the competition.

A Tip of the Top Hat

By the finals, seven teams were left to compete for top honors: AVHS, Kyushu, the US Naval Academy, Maryland, ETS, UTD, and Cornell. Robosub prides itself as a great spectator sport which may seem a little counterintuitive. It's a bit difficult to see the AUVs from the surface as they compete, but to remedy that organizers put on an absolutely fantastic webcast. During the finals, a panel including Daryl Davidson, Dr. Zoz Brooks from the Discovery Channel's Prototype This!, and a student representative from the competing team answered questions and gave color commentary. Cameras handled by divers and ROVs captured the action, and before runs there were team intro videos shot by sponsors and webcast organizers, 5:00 Films.

The final runs were a nail-biting affair with many teams having difficulty with some devilishly rearranged buoys. Teams like the US Naval Academy that were happy with their first run used the remainder of their 15 minutes to do a survey run — basically, a test run that would be used to collect data to improve the design for next year. The focus on the future was inspiring, and we are sure that it will keep the competition growing for years to come.

Cornell was the last team to go because they had the highest score going into the finals. Tachyon sailed through the hedges, followed the path, dropped markers, and shot torpedoes in what appeared to be a flawless run. Near the end, however, the AUV appeared struck with indecision as it hovered near the police station instead of surfacing with the counselor. Tachyon never ended up surfacing, making for a cliffhanger ending that could see a shakeup in the standings.

Awards were given out that evening. In addition to recognizing the highest scores, the University of Reykjavik received the Determination award, San Diego City Robotics was recognized for their outstanding group presentation, and the University of Central Florida received the highest scoring run for a team not in the finals. In the end, the top ranking were: Kyushu, 7th; UTD 6th; AVHS 5th; ETS 4th; Maryland 3rd; Naval Academy 2nd; and Cornell 1st — making it a repeat!

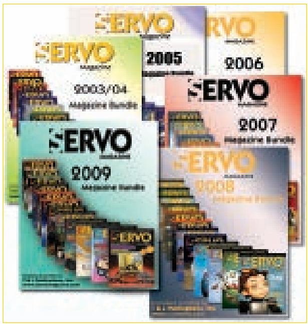
We would like to extend congratulations to all of the participants for turning in stellar performances in a very difficult competition. We're confident next year will see all of the same teams, some newcomers, and an even more challenging set of tasks. **SV**

RECOMMENDED WEBSITES

www.robosub.org

To order call 1-800-783-4624

SERVO Magazine Bundles



Published by T & L Publications, Inc.

$57 per bundle

Save $10 off the normal price!!

Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, and 09.

Call my Webstore and you'll get someone in AMERICA!

Visit my online store @ www.servomagazine.com

Kickin' Bot

by Grant Imahara

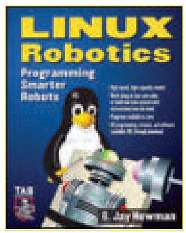
Enter the arena of the metal gladiators!

Do you have what it takes to build a battle-ready robot? You do now! Here are the plans, step-by-step directions, and expert advice that will put you in competition — while you have a heck of a lot of fun getting there. Grant Imahara, the creator of the popular BattleBot Deadblow, shares everything he's learned about robot design, tools, and techniques for metal working and the parts you need and where to get them.

$24.95

Linux Robotics

by D. Jay Newman

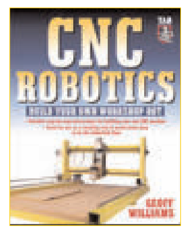


If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need.

$34.95

CNC Robotics

by Geoff Williams

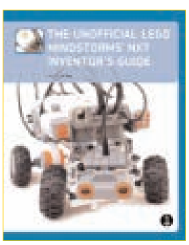


Here's the FIRST book to offer step-by-step guidelines that walk the reader through the entire process of building a CNC (Computer Numerical Control) machine from start to finish. Using inexpensive, off-the-shelf parts, readers can build CNC machines with true industrial shop applications such as machining, routing, and cutting — at a fraction of what it would cost to purchase one. Great for anyone who wants to automate a task in their home shop or small business. **$34.95**

SPECIAL OFFERS

The Unofficial LEGO MINDSTORMS NXT Inventor's Guide

by David J. Perdue



This book was written for the first version of the NXT set (#8527), and its projects are only compatible with the first version. In other words, because of piece differences between the NXT 1.0 and 2.0 sets, the projects in this book can only be built with an NXT 1.0 set. However, much of the other information is still helpful, and the building, mechanical, and programming details are still applicable.

Reg $29.95

Sale Price $25.95

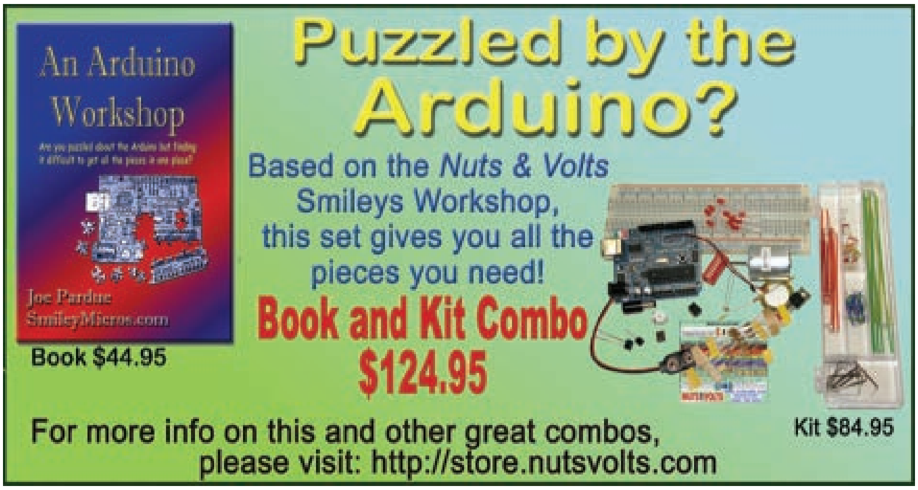
Technology Education Package for Everyone Starting in Electronics

This lab – from the good people at GSS Tech Ed – will show you 40 of the most simple and interesting experiments and lessons you have ever seen on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit. Along with the purchase of the lab, you will receive a special password to access the fantastic online interactive software to help you fully understand all the electronic principles. For a complete product description and sample software, please visit our webstore.

Regular Price $79.95 **Subscriber's Price $75.95**

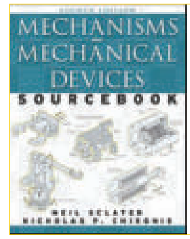
Or order online www.servomagazine.com

SPECIAL OFFERS



Mechanisms and Mechanical Devices Sourcebook

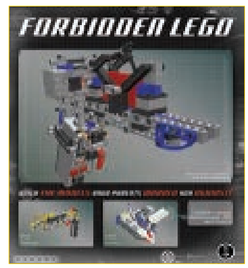
by Neil Sclater, Nicholas Chironis



Over 2,000 drawings make this sourcebook a gold mine of information for learning and innovating in mechanical design. Overviews of robotics, rapid prototyping, MEMS, and nanotechnology will get you up to speed on these cutting-edge technologies. Easy-to-read tutorial chapters on the basics of mechanisms and motion control will introduce those subjects to you. **Reg $89.95** **Sale Price $69.95**

Forbidden LEGO

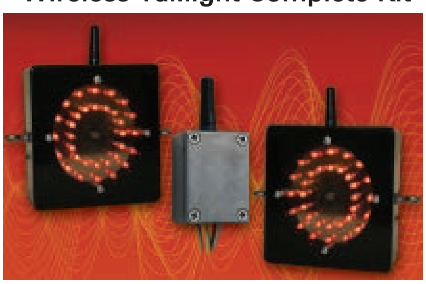
by Ulrik Pilegaard / Mike Dooley



Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse. **Reg $24.95** **Sale Price $19.95**

PROJECTS

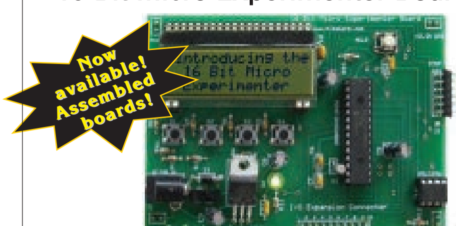
Wireless Taillight Complete Kit



As seen on the cover of the July 2010 issue of *Nuts & Volts*.
Never wire trailer lights again!
Just mount the taillights, plug in the transmitter, and you're ready to tow.
For complete kit details, please visit the webstore.
Includes an article reprint.

Subscriber's Price **$139.95**
Non-Subscriber's Price **$149.95**
A partial kit is also available.

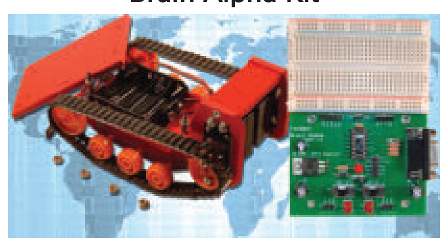
16-Bit Micro Experimenter Board



Ready to move on from eight-bit to 16-bit microcontrollers? Well, you're in luck!
In the December 2009 *Nuts & Volts* issue, you're introduced to the 16-Bit Micro Experimenter.
The kit comes with a CD-ROM that contains details on assembly, operation, as well as an assortment of ready-made applications. New applications will be added in upcoming months.

Subscriber's Price **$55.95**
Non-Subscriber's Price **$59.95**

Tankbot Kit & Brain Alpha Kit



Tankbot/Brain Alpha originally by Ron Hackett
Now with New Columnist Calvin Turzillo
A series filled with projects and experiments to challenge you through your learning process while you grow your fully expandable Brain Alpha PCB!
The brain is a PICAXE-14A!
For more info & pictures, visit the *SERVO* Webstore.
Tankbot and the Brain Alpha Kit can be purchased separately.

Combo Price $ 138.95

Then and NOW

ROBOT MANIPULATORS

by Tom Carroll

My wife and I recently returned from a vacation in Hawaii. One of the highlights was a trip on one of the 48-passenger Atlantis tourist submarines. Earlier, I happened to have met the senior pilot, Brian Ott of the Atlantis IX, based in Lahaina. He and his wife happened to be taking a short vacation at the same resort where we were staying. Sitting by one of the pools, we got to talking about his submarine. He knew that I had an interest in submarines and was 'into robotics'. We were discussing the underwater remotely operated vehicles (ROVs) that were trying to stop the Deepwater Horizon Gulf oil spill disaster and how they used their robot manipulator arms in various underwater tasks. Brian thought it might be interesting to have a pair of robot arms and 'hands' to pick up and examine various items he saw on the ocean floor, though tourist subs are certainly not research vehicles. He felt that a pair of arms working together like his own would work best, not just a robot arm as one would find on an industrial robot. During my trip in the sub, he allowed me to sit in the pilot's area with him and I could easily see how his view of the sea floor through the sub's large four foot diameter domed port (at the bow) made objects enticing enough to pick up and examine.

Sub Systems

Brian and his crew spent a few minutes showing me some of the submarine's different systems while there were no other tourists aboard. The Atlantis IX is part of the world's largest private submarine fleet. Each of the multi-million dollar vessels are meticulously built. **Figure 1** shows the Atlantis IX surfacing aft of the surface support vessel. I was amazed just how much of each subsystem was controlled in the same manner as mobile robots (especially high power combat robots). At 65 feet in length and displacing 80 tons, the sub runs on 20 huge 12 volt lead acid 1,280 amp hour batteries. The main bi-directional, brushless thrusters (two 20 HP main stern, two 10 HP vertical, and one 10 HP bow) and 20 1,000 watt exterior floodlights

FIGURE 1. Atlantis IX surfacing after a dive.

operate on 240 VDC. Main controls, an underwater telephone (to the surface support vessel), and emergency systems operate at 24 VDC. The Atlantis fleet also has a larger, 64 passenger sub with a 160 ton displacement.

With so much of the sub's control systems similar to mobile robot technology, we both felt that the addition of a pair of remote manipulator arms could easily be interfaced with the sub's control systems. These subs are more like an ROV that carries passengers than a larger military sub as they have multiple thrusters to move in three axes and rotation, plus they travel at slow speeds.

FIGURE 2. Space Shuttle Canadarm.

Robot Arms in Hostile or Inaccessible Environments

After talking with Brian, I got to thinking about robot manipulators as opposed to single industrial robot arms. People generally think of robot arms as being in the manufacturing arena, typically painting or welding cars. Industrial robots are rarely thought of as an extension of a human's arm and hand as they perform repetitive, programmed motions in a factory atmosphere. They are mainly envisioned as relieving humans of tedious, repetitive, boring, or dangerous work.

On the other hand, robot manipulators are usually thought of as extensions for a person's arms and hands and are typically teleoperated. Quite often they operate in pairs. The TV news coverage of the recent Gulf oil spill has shown ROVs with claws attached to rotating wrist joints, turning knobs on valves 5,000 feet down. An operator on the surface was instructing the ROV's arms and claws to perform tasks beyond the capability of a human diver. In contrast, the area in front of the Atlantis IX tourist sub was not particularly hostile at 125 feet, but was still well beyond the reach of the crew or the passengers. Sometimes it is this desire to handle or collect something that justifies the use of a remote manipulator.

Underwater Remotely Operated Vehicles

For those of you who may have read the piece, the July *SERVO* contained a great article by Dave Prochnow on the ROVs deployed in the Gulf to assist in the Deepwater Horizon disaster. At an ambient pressure of over 2,200 pounds per square inch at the depth of the blowout, typical manned diving operations were out of the question. To be useful in this difficult environment (gushing oil from a broken well head and blowout preventer), ROVs must have more capabilities than just lighting and a TV camera. Dual multi-axis arms with a selection of oil well specific claws are needed to manipulate various switches, valves, and debris, and deposit 'junk shot' into the wellhead blowout preventer. Operating under the sea can be extremely hazardous, but the oceans are not the only hostile place for robots (and humans).

Robot Manipulators in Space

One of the most famous robot manipulators in use today is the RMS or Remote Manipulator System used on the Space Shuttle (**Figure 2**). Back in 1985 when I was with Rockwell working on the NASA Space Station Project, I had the privilege of having astronaut Judith Resnik give me a short training session on the RMS simulator at NASA JSC. Handling multi-hundred million dollar payloads and even astronauts on the RMS' end-effector requires a reliable fail-safe and functional manipulator system. I later had to determine the robot manipulator system requirements for the Space Station which were similar, yet different from the shuttle's system.

These days, adding remote robot manipulators is fairly easy. However, the particular environment of the applications mentioned above can dramatically increase the complexity and expense required to design and build robotic manipulators. Going back about 65 years, one of the first applications for remote robot manipulators (or teleoperators) was in the early days of the Manhattan Project to produce the atomic bombs used in WWII. **Figure 3** shows a newer version of the same type of mechanical master/slave remote manipulators that were used to assemble radioactive bomb parts in the mid '40s. Mechanical linkages transferred the operator's motions directly to the claw and hand within the hot cell behind the radiation-proof glass window. Nowadays, robot manipulators have entered nuclear reactors to examine and repair radioactive parts, delicately grasping and

maneuvering items far too dangerous for a human to handle.

FIGURE 3. Master/slave manipulators.

Robot Manipulator Design Requirements

As I've mentioned, there are widely diverse scenarios that require robotic manipulation. Environments can range from a vacuum to extreme pressure to extreme cold to extreme heat. Earth's twin — the planet Venus — is not a particularly pleasant place for a space robot and certainly not for a human being. Venus is about as close to hell as anyone could ever imagine. With a surface pressure of almost as high as at the Deepwater Horizon site (at 1,323 pounds per square inch) and a temperature of 870 degrees Fahrenheit, if you or your spacecraft weren't squashed or burned, the sulfuric acid rains would certainly remove all of your skin in quick order.

In comparison, the bottom of the sea may not be as hot as the surface of Venus (unless an ROV is exploring a 'black smoker' volcanic vent) but the pressure can exceed 16,000 psi in extreme depths and actually be below freezing. Add in corrosive salt water that can ruin electrical and electronic systems in seconds and you have an extreme environment for deep sea explorations.

Designing a robot arm to just work on Earth is hard enough. Early robot arms frequently used hydraulic cylinders to provide enough force for the intended factory applications. Anyone who has ever used hydraulic systems invariably learns that they always manage to spring a leak at some time. Non-robotic uses such as heavy construction equipment that have booms, large road scraping blades, scoops, and similar motions that require a lot of force use high pressure hydraulic cylinders filled with oil. No matter how secure the seals, the oil under several thousands of psi of pressure leaks out and manages to collect dirt on every moving part. Early robots always seemed to be dirty.

Reversing the environment from thousands of internal psi trying to escape hydraulic cylinders, joints, and hoses, to extreme external ocean pressure trying to get into the mechanisms, electronics, and motors of deep sea robotic manipulators, it's clear you have a serious design hurdle. Designers of deep-diving ROVs with sophisticated manipulator arms have taken two different approaches to sealing the internal parts of the arms. One was to use hydraulic cylinders to move the arms and end-effectors in much the same manner as terrestrial applications (such as Caterpillar tractor blade cylinders). Some designs started to fail as external pressures actually were higher than the internal pressure, and salt water intrusion through failing seals caused corrosion and eventual failure.

The other approach was to forgo failure-prone high pressure seals and fill the interior with inert oil that is non-compressible, then use a flexible bladder to separate the sea water from the oil. This way, the interior of the motors and mechanisms are kept at the same ambient pressure as the outside ocean water. The oil-filled brushless AC or DC motors driving standard gear trains and mechanisms operate in the same manner as similar mechanisms on the surface. Rotating shaft seals do not require the same pressure integrity as they would under a much greater differential pressure. Electrical power and control lines still have to traverse from extreme pressure to minimal pressure, but these seals do not have moving components. Plus, they're easier to design and build. Lighting, certain types of

FIGURE 4. NASA Robonaut.

sensors, and TV systems still have to be constructed in a way to operate in the extreme pressure environment. At least the undersea vehicles do not have to use a diamond window to withstand the extreme temperature and pressure as did one of the Venus probes.

Does a Robot Manipulator Truly Mimic Human Anatomy?

To be of the greatest use to humans, a remote manipulator system should mimic most human hand and arm motions, and degrees of freedom. As humans, we like to say we're at the top of the chain of evolution because we have an opposable thumb and four fingers on each hand. We are tool users! However, to accomplish the extreme manual dexterity that we are so proud of, we possess over 135 degrees of freedom in our muscles, tendons, and over 80 multi-axis joints — far more than any robot ever produced. Degrees of freedom cost a lot of money, as any person who has ever built an 18 DOF humanoid or hexapod robot can attest to. Sometimes a designer may increase their robot's capabilities by giving one or more joints 360 degrees or more of rotation. This certainly helps if one wants to turn a valve handle, door knob, or similar object in both directions.

The NASA Robonaut

NASA developed the Robonaut shown in **Figure 4** to assist astronauts in space with difficult repair and assembly tasks. Lockheed Martin's Automation and Robotics group developed the wrist and forearm to closely mimic human movements. The robot has 7 DOF in each of its arms and has five-fingered dexterous hands containing a total of 10 DOF each with 2 DOF in the wrists. Robonaut also has 2 DOF in its head and waist. With five pounds of finger strength and force feedback in the finger tips, Robonaut can use slightly modified power and hand tools, and can actually pick up very small objects from a table — not really useful in the freefall space environment. Feedback from tactile sensors will soon allow the operator to 'feel' all types of objects.

Each arm can lift 21 pounds in 1G. The forearm contains 14 electric motors and 12 drive trains to move the fingers. There are two 3 DOF 'primary fingers,' two 1 DOF 'gripping fingers,' and the 3 DOF thumb working with a 1 DOF palm. The palm and its fingers can 'cup' an object much like a human. However, unlike a human's, this hand must be able to tolerate the vacuum of space, as well as the extreme temperature ranges.

The hand shown in **Figure 5** has 12 DOF: one 3 DOF in the thumb; two 3 DOF primary fingers; two 1 DOF gripping fingers; and one palm DOF. This makes Robonaut's hand a very dexterous 'tool' capable of performing many tasks while emulating a human hand. **Figure 6** shows the six linear actuators driving the digits and palm. The hand is divided into two functional sections: a dexterous work set which is used for manipulation; and a grasping set that allows the hand to maintain a grip on an object while manipulating it — a very essential feature for tool use.

The first two fingers and the thumb make up the dexterous work function. They have three degrees of freedom. They can open and close, as well as spread apart — essentially like human fingers. The remaining two fingers only open and close, and are used for grasping. The palm

FIGURE 5. Robonaut hand.

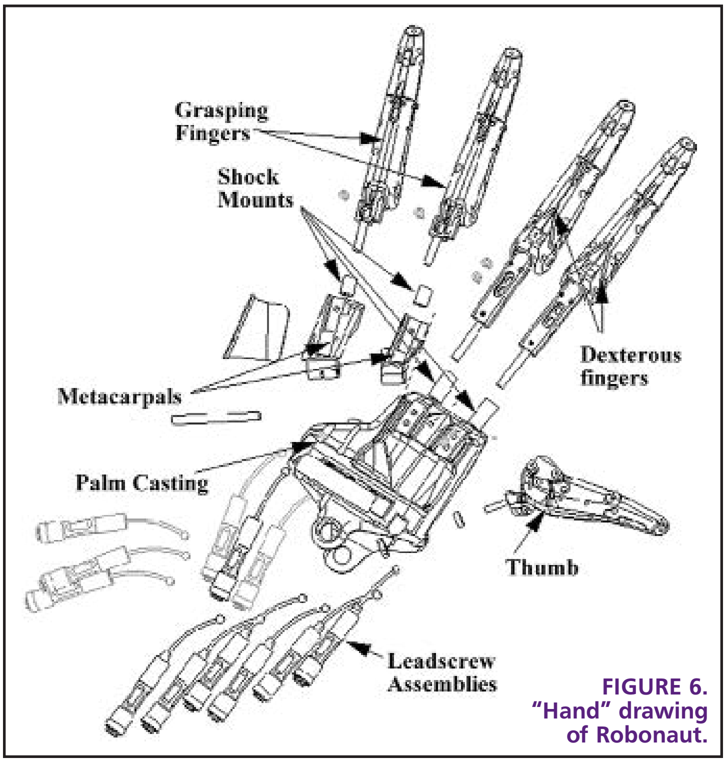


FIGURE 6. "Hand" drawing of Robonaut.

can cup to help grasp tools, too. It had to be functional enough to work with a variety of tools and other devices, and rugged enough to survive the space environment. Plus, it had to be extremely compact (much like a human hand, wrist, and arm).

The Willow Garage PR2

The PR2 from Willow Garage in Menlo Park, CA is a most unique robot. The drawing in **Figure 7** shows just how much of the volume of the PR2 is dedicated to the arm structures. You might notice that the PR2 does not have multi-fingered hands like the Robonaut but instead uses a more standard robot parallel jaw claw. This 'hand' design doesn't make the robot any less functional for its intended uses. On a recent visit to Willow Garage, I managed to get some close-up views of PR2's arm — truly the most unique humanoid robot design I've ever seen. Instead of the typical shoulder joint protruding from the top of a chest cavity, Willow Garage employed two rotating cylindrical structures on each side of the robot upon which the arm shoulder joints and arm assemblies are mounted. **Figure 8** shows the view looking downward at the top of a right arm. The parallel jaw claw is visible as well as the beautifully machined weight compensating mechanisms.

The PR2 has two 7 DOF arms with a range of motion similar to that of a human to pick up and manipulate common household objects. Each arm has pan and tilt shoulder joints, upper arm roll joints, elbow flexion joints, forearm roll joints, wrist flexion joints, and gripper roll joints. Each arm carries a single DOF gripper capable of grasping most small objects up to 1.7 Kg, and is outfitted with a pressure sensor array on the gripper tips. The arm's four main DOFs are back-drivable so when the arm encounters an immovable object, the force of the interaction drives the motors back and the arm gives way. This allows the robot to operate in unstructured environments. The spine of the PR2 can telescope upward to extend its reach from objects on the ground to objects on countertops.

Final Thoughts

Robot manipulators have come a long way since the early Oak Ridge hot cell robot arms. Industrial robot manufacturers have seen the need to extend their product line into multi-arm, multi-axis manipulator systems such as the new Yaskawa Motoman SDA5D shown playing with some LEGO blocks in **Figure 9**. There are many applications where two multi-axis arms are far better than a standard industrial robot arm. Factory assembly lines, space rovers, undersea ROVs, tourist submarines, and robots in the home will all soon have manual dexterity and the manipulative ability of humans. **SV**

Tom Carroll can be reached at TWCarroll@aol.com.

FIGURE 7. PR2 drawing.

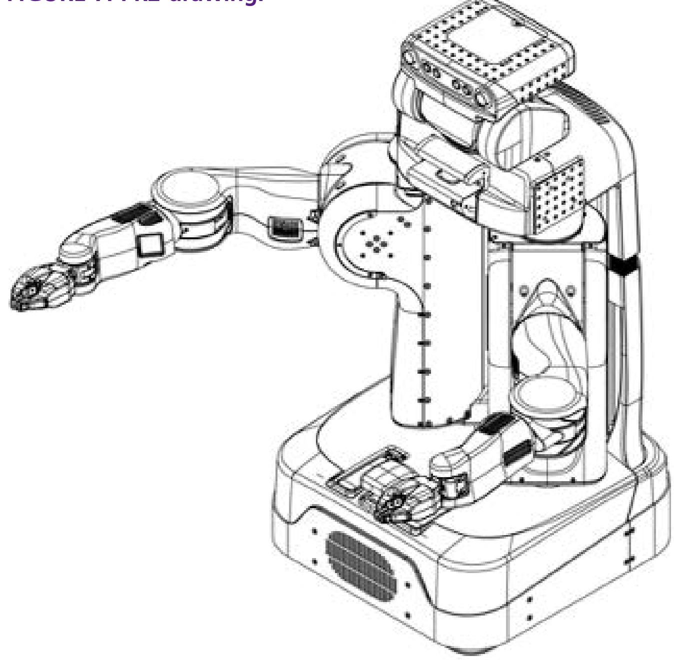
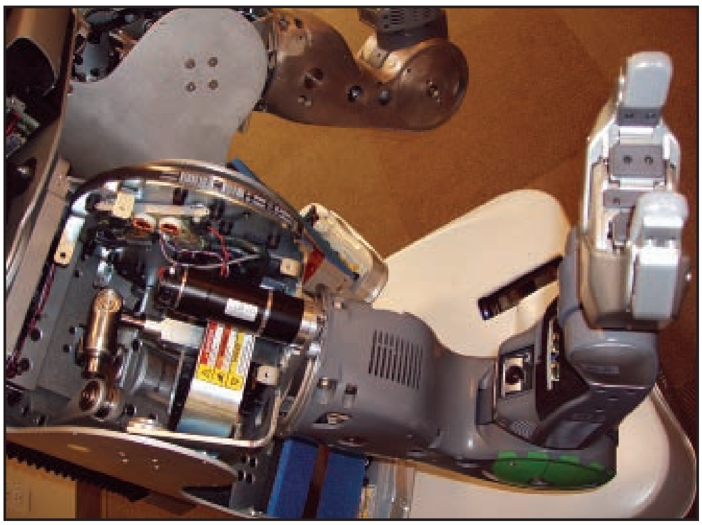


FIGURE 8. Top of PR2's arm.



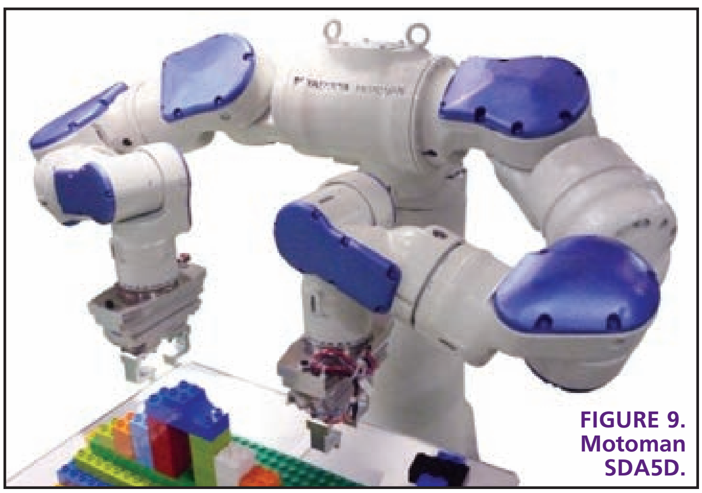


FIGURE 9. Motoman SDA5D.

ROBO-LINKS

For the finest in robots, parts, and services, go to www.servomagazine.com and click on Robo-Links.

To Advertise: Call 951-371-8497

Advertiser Index

All Electronics Corp. 21, 82
AP Circuits 39
BaneBots 7
GSS Tech Ed 46
Linx Technologies 82
Lynxmotion, Inc. 2
Maxbotix 82
PCB Pool 19, 82
Pololu Robotics & Electronics ..10, 82
RobotShop, Inc 3, 82
SchmartBoard 83
Solarbotics/HVW 7
Sunstone Circuits ....Back Cover
Technological Arts 21
Vantec 19


NO MATTER WHAT THE IDEA
YOUR PCB PROTOTYPES SHOULD BE THE EASY PART
QUOTE & ORDER PCBS ONLINE AT WWW.SUNSTONE.COM OR CALL 1-800-228-8198